ISBN 978-82-326-4080-5 (printed version)
ISBN 978-82-326-4081-2 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Doctoral Thesis

Patrick Schuch

Doctoral theses at NTNU, 2019:242

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Information Security and Communication
Technology

NTNU

Doctoral theses at NTNU, 2019:242

Patrick Schuch

# Deep Learning for Fingerprint Recognition Systems

NTNU
Norwegian University of
Science and Technology

Patrick Schuch

# Deep Learning for Fingerprint Recognition Systems

Thesis for the degree of Philosophiae Doctor

Gjøvik, October 2019

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Information Security and Communication
Technology

NTNU
Norwegian University of
Science and Technology

**NTNU**
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Information Security and Communication
Technology

ISBN 978-82-326-4080-5 (printed version)
ISBN 978-82-326-4081-2 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:242

Printed by Skipnes Kommunikasjon as

# Declaration of authorship

I, Patrick Schuch, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Patrick Schuch)

Date:

# Summary

Biometric recognition is a typical means to identify individuals or to verify claimed identities. Use cases are manifold. For example, users can unlock their smartphones for convenience by presenting their faces or fingerprints. Or one's identity is verified when crossing borders. Today, biometric recognition already has many points of contact with our daily life and there are more to come.

Besides iris and face, fingerprint is the most wide spread biometric trait used for recognition. Fingerprints are assumed to be unique for each and every finger. This makes it an ideal trait for recognition. In addition, fingerprint recognition has more than a century of tradition in the field of biometric recognition. A great amount of expertise and engineering skill made it a quite mature technology over time. Only few false positive and false negative errors are made in recognition in today's deployed systems.

However, fingerprint recognition is still far from being perfect. In contrast to popular opinion, fingerprint recognition is not a solved problem. Actually, there is still a lot of work to do. As biometric systems become larger and become more inclusive, even new challenges arise. Systems need to deal with large amounts of data while keeping performance with respect to recognition performance as well as transaction times in a reasonable order. Recognition shall work for everyone and shall not exclude a certain ethnic group or subset of the population. It will work in unconstrained conditions. However, it shall still make no erroneous decisions. Engineering may have come to its limits at this stage.

In contrast to classical engineering, machine learning based on artificial neural networks may be a reasonable alternative. The emerging technologies of Deep Learning achieve tremendous successes in many domains of image processing and pattern recognition. This work assesses the application of such innovative machine learning concepts to fingerprint recognition. Three central aspects and challenges in fingerprint recognition are inspected in detail: fingerprint sample enhancement, orientation field estimation, and efficient processing structures.

# Sammendrag

Biometrisk gjenkjenning er en vanlig måte brukt for å identifisere enkeltpersoner eller for å verifisere en hevdet identitet. Bruksområdene er mangfoldige. For eksempel kan brukere låse opp smarttelefoner på en enkel må te ved å presentere deres ansikt eller fingeravtrykk. Et annet eksempel er når ens identitet blir verifisert når man krysser en landegrense. I dag kommer vi i kontakt med biometrisk gjenkjenning flere steder i det daglige liv, og det vil komme flere fremover.

Utenom iris og ansikt, er fingeravtrykk det biometriske kjennetegnet som er mest vanlig når det kommer til gjenkjenning. Fingeravtrykk er antatt å være unik for hver enkelt finger. Dette gjør det til et ideelt kjennetegn for gjenkjenning. I tillegg har fingeravtrykk mer enn hundre års tradisjon innenfor biometrisk gjenkjenning. En stor andel ekspertise og teknisk utvikling har gjort det til en ganske moden teknologi over tid. Kun noen få falske positive og falske negative feil blir gjort i systemer som er tatt i bruk i dag.

Derimot er gjenkjenning av fingeravtrykk fremdeles langt fra å være perfekt. I motsetning til hva folk flest tror, er ikke gjenkjenning basert på fingeravtrykk et løst problem. Det er fremdeles mye arbeid som gjenstår. Etter hvert som biometriske systemer blir større og mer inklusive oppstår det nye utfordringer. Systemer må håndtere store mengder data samtidig som ytelse når det gjelder gjenkjenning og transaksjonstider opprettholdes på et rimelig nivå. Gjenkjenning skal virke for alle og skal ikke ekskludere basert på etnisitet eller undergrupper i befolkningen. Det skal virke under alle forhold. Likevel, skal det ikke gjøres noen feilaktige beslutninger. Den tekniske utviklingen kan ha nådd sine begrensninger på dette stadiet.

I motsetning til klassisk ingeniørvitenskap kan maskinlæring basert på kunstige nevrale nettverk være et rimelig alternativ. Den fremvoksende teknologien innenfor dyp læring oppnår fremragende suksess innenfor flere områder av bildeprosessering og mønster gjenkjenning. Dette arbeidet undersøker anvendelsen av slike innovative maskinlæringskonsepter på gjenkjenning av fingeravtrykk. Tre sentrale aspekter og utfordringer innenfor fingeravtrykk gjenkjenning blir vurdert i detalj: Forbedring av fingeravtrykkprøver, estimering av orienteringsfelt, og effektive prosesseringsstrukturer.

# Acknowledgements

I would like to gratefully thank my supervisor at NTNU, Prof. Dr. Christoph Busch. He was a great guide through my studies. Despite his tremendous workload, he is pure commitment. He is always openminded and a source of ideas and inspirations. In addition, he is an extraordinary networker. Thanks to Prof. Dr. Katrin Franke for her co-supervision.

Representative for many more at NTNU, I want to thank Raghu, Kiran, Patrick, Martin, Guoqiang, and Pankay for sharing their ideas and inspirations in all those coffee breaks. Special thanks to Hilde, who knows a solution to almost any problem.

This work would not have been possible at all without my employer: DERMALOG Identification Systems GmbH. I wish to express my deepest gratitude for offering me this great opportunity. This would have not been possible without our CEO Günther Mull. I had two remarkable brilliant supervisors at DERMALOG: Simon and Marek. Both are fantastic sources of ideas and were always willing to go the extra mile with me. Thanks to Simon, Marek, David, Dirk, Daniel, and Marcus for all the ideas, creativity, support, laughter, and way too much coffee.

Thanks to my friend John for endless night shifts of proofreading.

A big thank you goes out to my three sons, Alexander, Daniel, and Tobias, for continuously recharging me in my spare time and focussing me on what really matters in life. Thanks to my parents, Monika and Dieter, simply for being there.

Last but not least, I want to thank my wonderful wife, Kathrin, for her willingness to start this enterprise with me in the first place. Her everlasting support is just priceless. Going all the way together with her was an amazing experience.

# Contents

# List of Tables

# List of Figures

# Part I

# Deep Learning for Fingerprint Recognition Systems

# Chapter 1

# Introduction

## 1.1 Machine Learning

### 1.1.1 General Concepts

Machine learning is a discipline in computer science, which deals with making predictions based on given information. A *machine* can actually *learn* to solve tasks like making *decisions*, *estimations*, and *predictions*. Sophisticated *learning* or *training strategies* exist and are applied to obtain a prediction.

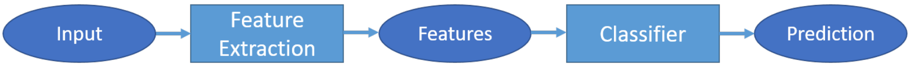


**Figure 1.1:** In the classical workflow of pattern recognition, features are extracted by a feature extraction process first. A classifier predicts from these features.

In this context, a *classifier* or *model* $\mathcal{M}$ can be any algorithm, which generates a prediction $y$ from an input $x$ of data. Depending on the context, a single input data $x$ may also called a *sample*, an *observation*, *features*, or just short *data*. Likewise, a prediction $y$ may be called an *estimation* or a *decision*. The model $\mathcal{M}$ is therefore a function of the input data $x$:

$$\mathcal{M} : x \mapsto y \tag{1.1}$$

A model is called *trainable*, if the model $\mathcal{M}$ depends on parameters $\Theta$, which can be optimized with respect to an *optimization criterion*. The model $\mathcal{M}$ adjusts its predictions according to the data it has experienced so far.

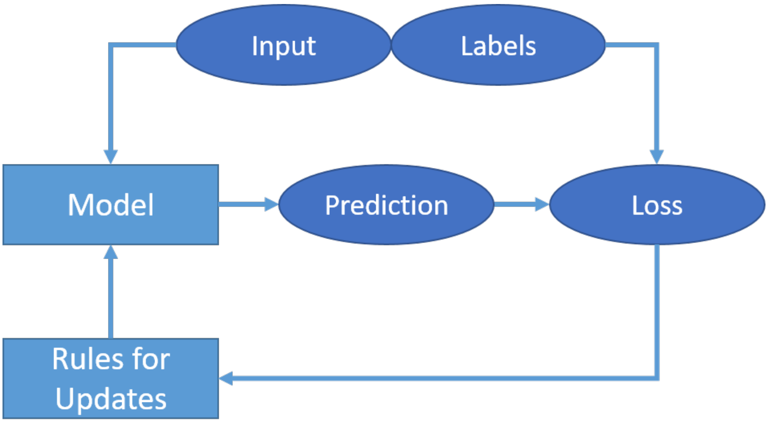


**Figure 1.2:** A training strategy consists of three parts: presentation, loss function, and parameter update. First, input is presented in a defined manner to a trainable model $\mathcal{M}$. Then, the model generates predictions based on the input and its internal parameters $\Theta$. The loss function generates loss by evaluation of the predictions (optionally using label data). The loss can be used to update the parameters $\Theta$.

The training of a model $\mathcal{M}$ is driven by the input data that is learnt and by the training strategy that is applied. A training strategy consists of three parts: a *loss function*, rules for *optimization* of the parameters $\Theta$, and a protocol for the *presentation* of training data (see Figure 1.2).

A *loss function* $l$ is a function, that assigns a value of *loss* to a prediction given a specific input data. The loss function may also be called a *cost function* and the loss may be called a *cost* respectively. The loss function represents the optimization criterion, which is a formal expression for the task, that the model $\mathcal{M}$ shall learn: If the model $\mathcal{M}$ reacts with the correct decision to a given input, a low loss is assigned to this decision. However, if the model $\mathcal{M}$ makes incorrect predictions, higher losses are assigned. A loss function can consist of several partial losses. Each partial loss may represent a single optimization criterion. Training a model $\mathcal{M}$ shall reduce the loss function $l$ of the model's decisions $\mathcal{M}(x, \Theta)$ for an input $x$ and not necessarily required expected decision $\hat{y}$:

$$\min_{\Theta} l(\mathcal{M}(x, \Theta), \hat{y}) \tag{1.2}$$

Training a model $\mathcal{M}$ is therefore an optimization problem. It is the optimization of the adjustable parameters $\Theta$ within the model. In each optimization step, the parameters are *updated* according to what is currently learned. It is beneficial, if the loss function is differentiable with respect to the adjustable parameters $\Theta$ within the model $\mathcal{M}$. Differentiability allows to modify the model $\mathcal{M}$ in a systematic and analytically reasonable manner. For example, it allows to apply a *gradient descent* approach leading to a minimized overall error..

Another important aspect of the training strategy is the way, in which training data is presented to the model $\mathcal{M}$. In some strategies single samples are presented to a model $\mathcal{M}$ in each training step. In other cases, sets of samples are presented to the model $\mathcal{M}$, e.g. two samples at a time in *Siamese Networks*[34], three samples in *Triplet Networks*[109], or sets of several samples like in *SphereFace*[193]. The model $\mathcal{M}$ then learns from these sets of samples, whether the they belong to the same class.

If the model $\mathcal{M}$ shall learn the relation between samples, the training strategy requires such information on the relation. In case of biometrics, such an information on relation may be whether two samples are mated meaning they stem from the same source, i.e. the same biometric instance. Such information is often called a *label*. But label information is not available in all cases. Thus, two types of training can be distinguished: *supervised* and *unsupervised* training.[1]

For the former additional information is present and used during training. Generation of such label data can be expensive. Large amounts of labels are necessary for successful training and reasonable testing of a trained model. In addition, the labels have to be reliable to be useful.

For the latter, no additional information is required. Unsupervised training allows to use any reasonable data without the need for labels. It is often used to discover the relevant structure from the data, e.g. in *Auto-Encoders* [15].

The most common use case for the usage of label data is, that a model $\mathcal{M}$ shall learn to predict the label from the input data. There are two categories of such predictions: *classification* and *regression*. The former is used to predict discrete values or classes. The latter is used for the prediction of continuous values.

A typical challenge in machine learning is a lack of *generalization* capabilities of trained models: This is when trained models performs well on training data, i.e. the data they already know, but fail when processing unknown test data. If this is the case, the model is *over-fitted* to the training data. This may be due to systematic differences between training data and test data. This difference is known as the *co-variate shift* or *dataset bias*. But this also may be due to the fact, that the model has learned the training data completely by heart. In general, every

[1] There are also hybrid versions of training, e.g. where only partial information is available. These trainings are called *semi-supervised*.

trainable parameter is actually a degree of freedom for a model. If a model has too many degrees of freedom, it is likely to happen, that the model will over-fit to the training data. However, being unable to deal with unknown data is critical, since the model may fail in any environment, which is different from the training environment. Special techniques have to be applied to enable a model's ability to generalize well to unknown data.

### 1.1.2 Hand-crafted Features versus Learning Features

The base of every decision are the features. The features need to be relevant for the value to be predicted. In the domain of digital image processing, the obvious input into a machine learning approach is an image. Usually, the images are not processed directly. The features used for prediction need to be *extracted* from the images first. One can distinguish between two types of features:

- *Hand-crafted* or *engineered* features
- *Learned* features

In classical, hand-crafted approaches, engineers design features that are extracted from the images. Usually, such features are chosen by the engineers, because those features seem to be relevant for solving a given task. A feature extraction approach relies on the expertise of its engineers. Thus, these features are understandable by a human. Such an interpretability might be required for transparency in automatic reasoning, e.g. when applying forensic testing or the *General Data Protection Regulation* of the European Union.

But is also possible to learn features in a data driven manner. In this case, statistical and systematic characteristics are used as features. Commonly, data driven approaches learn to extract helpful features, which are not of obvious relevance for a given task. Thus, those features are more difficult to interpret compared to traditional handcrafted features. An algorithm might extract features, which might have no obvious relevance for a given task. In fact, the actual features might be anything, what helps solving the given task. In contrast to engineered features, learned features may be hardly understandable to a human in some cases. However, due to the analysis of the data, the data-driven approaches are likely to be more effective than the engineered ones.

### 1.1.3 Neural Networks

*Artificial Neural Networks* (ANN) are a class of well known classifiers or models, which are inspired by a simplification of the behaviour of real neurons (see Figure 1.3). This simplified behavior can be summarized as follows: An ANN is *stimulated* by external *stimuli*, i.e. the input data. Each neuron in the first layer of the ANN is connected to all external stimuli. The external stimuli usually features extracted from the original input data (see Figure 1.4). The values of the stimuli

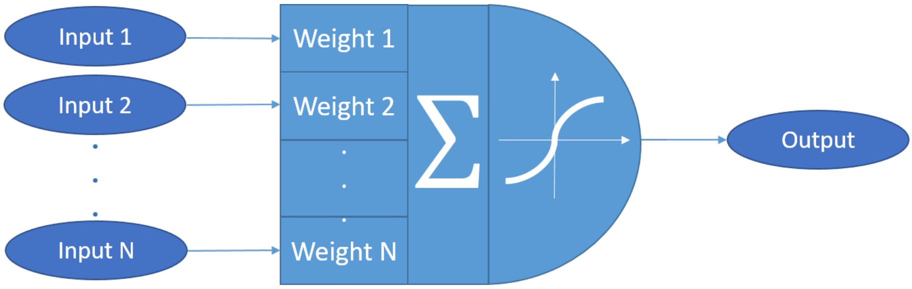


**Figure 1.3:** In an ANN a single neuron generates a weighted sum of all inputs. The weighted sum is propagated through an activation function. The result is the output of the neuron. This output may be used as the input to another neuron.

are multiplied by factors within the neuron. The value of a single factor represents the sensibility of the neuron to a specific input stimulus. These factors are called *weights* and form the set of trainable parameters $\Theta$ of an ANN. The count of trainable parameters usually ranged from less than a hundred for more easy tasks up to a few tens of thousands for tasks like face recognition [168]. Actually, the weights are adjustable. Thus, the sensibility of a neuron to a stimulus is trainable. The weighted input stimuli are summed up. If this weighted sum of stimuli exceeds a specific level of stimulation, the neuron generates a new stimulus. A non-linear *activation function* is commonly used to represent this specific level. The commonly used activation function for ANNs was the *Sigmoid* function $\sigma$:

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \tag{1.3}$$

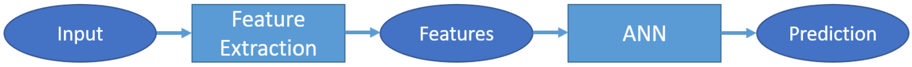


**Figure 1.4:** Artificial neural networks (ANN) are a class of classifiers, which can predict from features (compare to Figure 1.1).

In ANNs neurons are usually organized in sets. In general, each set of neurons shares and processes the same input. These sets are called *layers*. The output stimuli of all neurons of one layer can be used as the input to the next layer. The layer dealing with the input data is usually called *input layer*. The last layer is called *output layer*. All layers between the input and the output layer are called *hidden layers*. Figure 1.5 visualizes the architecture of an ANN.

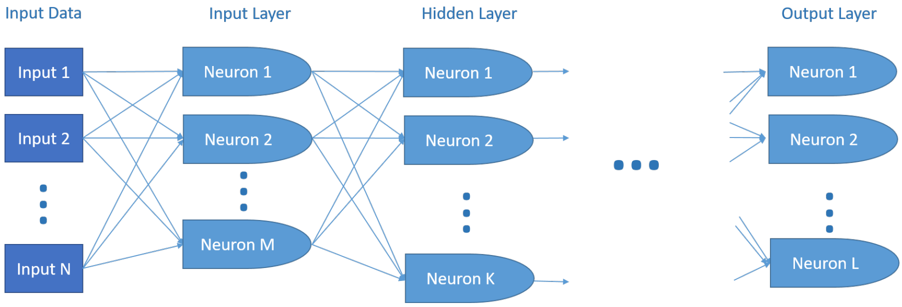


**Figure 1.5:** An ANN can consist of an arbitrary number of layers. Each layer consists of an arbitrary number of neurons. The input layer deals with the input features. The last layer is the output layer. All layers inbetween are called hidden layers. The output of each layer can actually be the input to the next layer.

The Sigmoid function (see Equation 1.3) has some beneficial attributes. The function is smooth and it maps the entire range of real values to the range $[0, 1]$. More important is the fact, that the Sigmoid function can be differentiated very easily:

$$\frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} = \sigma(x)(1 - \sigma(x)) \tag{1.4}$$

This keeps the effort required for differentiation quite low and allows to differentiate the loss function with respect to the weights of the output layer. Figure 1.6 visualizes the mapping and the derivative of the Sigmoid finction. The input of the output layer is the output of the layer before, which allows to differentiate the loss in each layer with respect to the weights in the layer before. By application of the chain rule for differentiation, the gradients of the loss function can be computed for all parameters $\Theta$ within the ANN. This technique is called *Error Back-Propagation*. It allows to apply a gradient descent for optimization on the model's weights $\Theta$.

There is a solid theoretical foundation for ANNs [71]. An ANN can have a large *expressive capacity*, i.e. how many function it can approximate. It can be shown, that an ANN with at least one hidden layer can approximate any given target func-

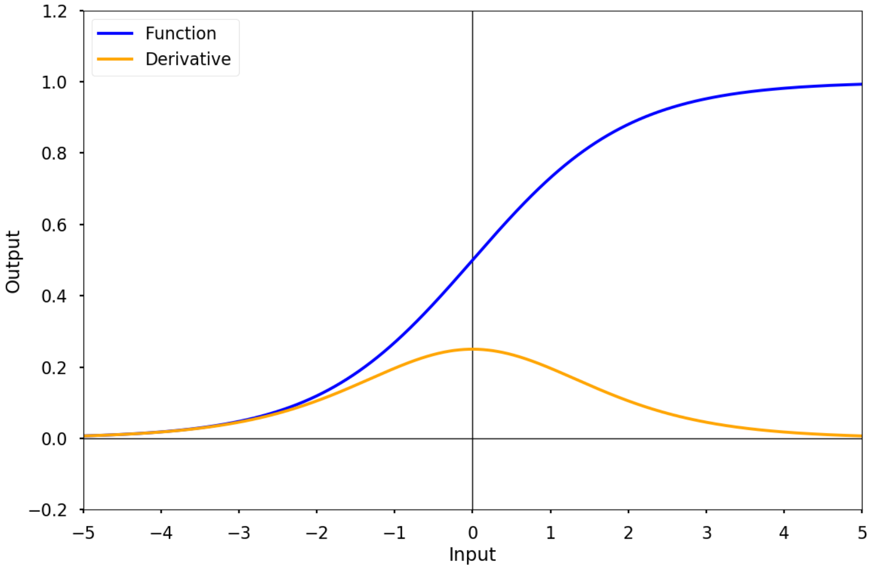


**Figure 1.6:** Classical ANNs used the Sigmoid function as transfer function. The Sigmoid function is easily derivable.

tion, if the number of neurons in the layers is sufficiently large. They shall be able to solve any problem at least theoretically.

The influence of the number of layers on the capacity is stronger than the influence the number of neurons in each layer. The non-linearity of the activation functions between the layers causes the expressive capacity of a model to grow exponentially in the number of layers. Thus, one would prefer to use more layers over the usage of more neurons per layer. Unfortunately, the Sigmoid function has an undesired attribute: It damps the error which is propagated back significantly from each layer to the next. The gradients calculated for each layer decreases the more layers they are propagated to the front of the ANN. This phenomenon is called *vanishing gradients*. It makes it hardly possible to update parameters in first layers of an ANN.

There was no solution to the problem of vanishing gradients. Hence, *shallow* networks of few *broad* layers with large numbers of neurons were used until Deep Learning overcame this problem (see 1.1.4).

Application to images was difficult, too: The straight forward approach would be to treat each pixel in an image as a single feature. This would of course result

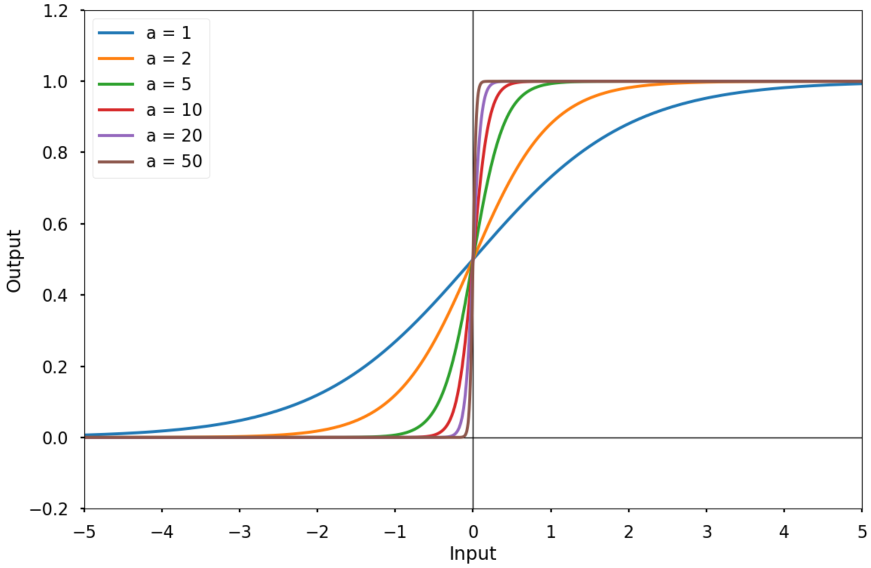


**Figure 1.7:** The family of functions $\sigma_a(x) = 1/(1 + (\exp(-ax)))$ can be tuned by a slope parameter $a$. It can be used to approximate the unit step function while being still differentiable.

in very large input feature vectors. Such a processing was neither reasonable nor feasible. Development in the domain of neural networks slowed down. Other new and powerful approaches in machine learning came up. *Support Vector Machines* (SVM)[59] became state of the art for a long time, e.g. at the ImageNet benchmark [249].

### 1.1.4 Convolutional Neural Networks

However, development and research in the domain of neural networks never died entirely. *Convolutional Neural Networks* (CNN) came up with some brilliant ideas, which allowed application to images and circumvented the known limitations in ANNs [171]. CNNs paved the way for *Deep Learning* (DL).

The central idea for application to images is to change the way a neuron processes a given input. A neuron in a CNN works like a digital filter on the input. The output of a neuron is then a feature map rather than a single value. Such a filter can detect a feature independent from the position of the feature. It keeps local coherence in the output data, where casual ANNs would loose the coherence, because they treat the data as vectors (see Figure 1.8). It also reduces the numbers of parameters significantly while keeping the capacity of a model.

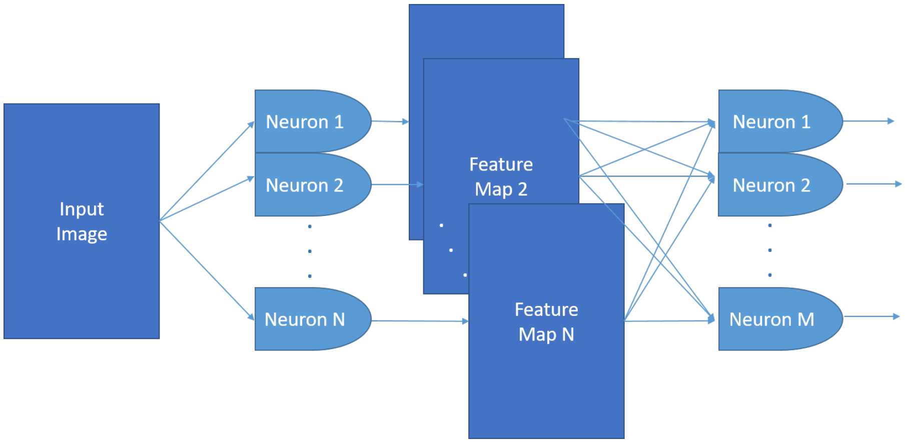


**Figure 1.8:** In a CNN, neurons generate feature maps. Like in ANNs the output of a single layer can be used as an input to another layer. This allows to recombine features. While usually simple features are learned in the first layers of a CNN, features in later layers may be very complex. This allows a remarkably large capacity of CNNs.

In addition, the activation function was changed from the Sigmoid function to a *Rectified Linear Unit* (ReLU):

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{1.5}$$

This change may seem very little. Actually, it is an important change. The most important attribute of an activation function is its non-linearity. ReLUs provide the required non-linearity and allow an extremely simple differentiation, while not damping the gradients.

$$\frac{\mathrm{d}\,\mathrm{ReLU}(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x < 0 \end{cases} \text{and define } \left.\frac{\mathrm{d}\,\mathrm{ReLU}(x)}{\mathrm{d}x}\right|_{x=0} = 0 \tag{1.6}$$

The problem of vanishing gradients therefore was attenuated. Figure 1.9 visualizes the mapping and the derivative of the ReLU function.

This in turn enabled the use of *deep* networks consisting of almost arbitrary numbers of layers, e.g. ResNet contains more than thousand layers with a total of 19.4 million parameters [107]. The number of trainable parameters in modern CNNs is therefore by orders of magnitude larger than it used to be in ANNs. For constant number of parameters, the capacity of deep models is significantly larger than the

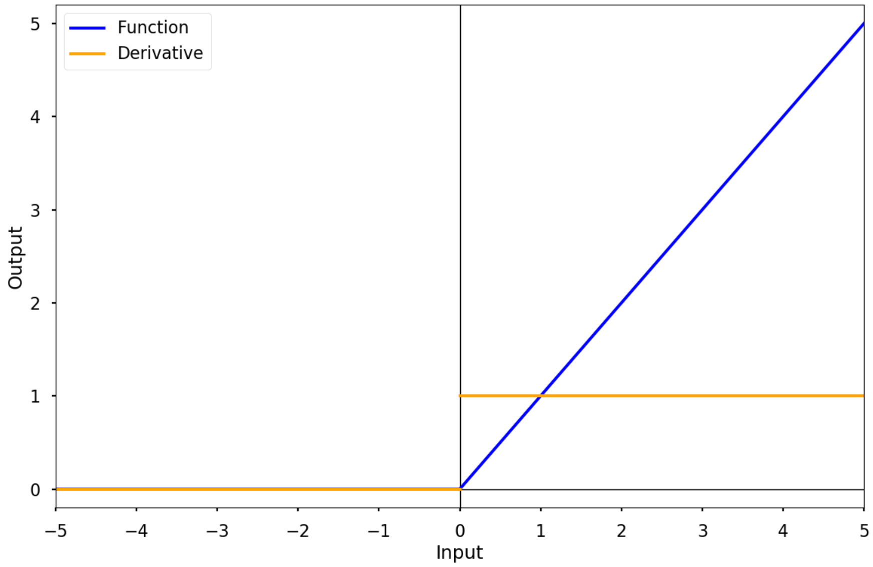


**Figure 1.9:** CNNs used the ReLU function as transfer function. Compared to the Sigmoid function, it is derivable even more easily. It also evades the problem of vanishing gradients..

capacity of shallow networks. Using many layers enables to recombine the feature maps of single layers. By doing so, simple features in the early layers can be recombined to more complex features in the later layers.

In the early days of CNNs, the parameters of the models were initialized by sophisticated strategies [72]. For example, appropriately dimensioned Auto-Encoders were trained in an unsupervised manner. This allowed to train layers, which can describe the data in a reasonable manner. The trained layers were then used as the layers of the model dealing with the actual problem. As the layers are already able to represent the data reasonably, this initialization improved the convergence of the training on the actual problem. Nowadays, such an initialization is not necessary anymore and is therefore only seldom used, because models usually converge fast enough.

These ideas coincided with two significant changes in the environment of machine learning. First, large amounts of usable data became available with the steady rising of the internet. For example, the vast amount of face images in Facebook allowed training of a face recognition. This has been allowing to train models with large amounts of trainable parameters. Second, *Graphical Processing Units*

(GPU) came up, which allowed fast and highly parallel processing of such large amounts of data. Both factors paved the way for CNNs in the first place.

Another important factor in the rise of DL is the commercial interest in the field of data mining. Dealing with large amounts of information is an important economic sector. Thus, global players like Google and Facebook fostered the development and research in this field. They even share parts of their research and development with the public and provide necessary technology and infrastructure, e.g. the DL framework *tensorflow* [1].

Last but not least, the structure of CNNs is actually a paradigm shift. In classical approaches, feature extraction and pattern recognition are separated: First, features are extracted from the data. Second, a decision is made based on the features. Both aspects were improved and optimized separately. In contrast, one simply cannot separate both aspects in CNNs (compare Figures 1.1,1.4, and 1.10). It is not clear in which layer the feature extraction ends and the decision making starts. Thus, both aspects are trained jointly. This holistic approach allows to learn to extract features, that actually help making the correct decision.

Input
CNN
Prediction

**Figure 1.10:** The workflow of a CNN is different to the one of the classical approach (compare to Figure 1.1). CNNs do not clearly split feature extraction and decision making. This allows to train a feature extraction, which is helpful for solving the actual task.

Then in 2012, CNNs finally had their breakthrough at the famous ImageNet benchmark [163]. CNNs significantly improved the performance over the former state of the art [248]. They suddenly became state of the art themselves. Today, almost all benchmarks in the domain of image processing and pattern recognition are dominated by CNNs. Tremendous improvements in performance were achieved over the last years in terms of accuracy as well as speed [242]. Nowadays, CNNs even outperform human performance, e.g. playing the game Go [90].

Meanwhile, there are DL frameworks and tutorials freely available, which allow an easy access to this powerful technology, e.g. Tensorflow[1], theano[284], PyTorch[232], keras[58], or caffe[141]. This is boon and bane of this development. On the one hand, anyone capable of minimal programming skills is able to use DL. It enables everyone to do research with this powerful tool. On the other hand, it might be even too easy to use. Even in academia DL often seems to be applied without any understanding of DL itself. Reproducibility of reported results in research is seldom given. In some cases, the researchers even admit, they cannot repeat their own experiments, e.g. the famous GoogLeNet [278]. Little assessment of consequences of the results produced is given.

Improvements in GPU technologies and DL strategies led to very deep models. Models with more than a thousand layers were trained successfully [107]. Such models achieve impressive results. However, this depth makes what is happening inside a model hardly understandable. Deep networks need to be treated as black boxes.

This lack of understanding is only one of the current challenges in DL. There are several further critical challenges, that stand against the great successes of DL.

While there is a solid theoretical foundation for ANNs, there is no such foundation for CNNs. We are only about to start getting an idea, what is happening exactly inside the model. We also do not know, why it is happening. CNNs therefore achieve only unreasonable successes so far. However, the DL community is aware of the problem and tries to improve on both aspects: understanding of the theory and understanding of the actual models [93]. Another branch of science in this domain is concerned with *explainable artificial intelligence* [99]. This branch of research may also introduce a feedback to hand-crafted feature design: Learning and understanding of the successes may be used to improve hand-crafting and engineering of features.

CNNs can be very sensitive to the input data. Actually, the ill position of the problem resulted in the effect, that small changes in the input data can force the models to fail. For examples, *adversarial samples* were designed by minimal distortions of actual data with the aim to disturb a trained model [277]. There are adversarial samples with distortion so small, that they are not perceivable, i.e. below the quantization level for discrete valued images. Such samples show, that the processing models are very sensitive to the input data. This is of course also an indicator for overfitting. Today, research works on this challenge. *Generative Adversarial Networks* (GAN) are used to generate samples from the data already learned [92]. Such samples can in turn be used to improve the stability of a model.

Another obvious challenge is the fact, that models grew larger and larger over time. While smaller networks are already complex, larger networks are even more complex. This actually is a problem in many ways. Most importantly, too large networks are simply not deployable on systems with limited resources. There is a branch of research in the domain of DL, that deals with improvements on the trade off between capacity and resources [113].

Another important issue in the near future will be the usage of unsupervised techniques. Unsupervised learning can be seen as understanding the data by the data itself. Not being dependent on labelled data will enable further improvements.

Last but not least, another open issue so far seems to be the amount of data used for training. For many problems only little data is available. Learning and generalization from very little sample data is one of the most impressive effort of our human brains. This ability represents the understanding of *concepts* rather than just making good predictions. To my mind, the most import challenge in the next years will be dealing with little data. I assume, that solving this challenge will again enable ground breaking improvements for DL.

## 1.2 Fingerprint Recognition and its Challenges

### 1.2.1 Biometric Recognition

*Biometric recognition* is defined as the "*automated recognition of individuals based on their behavioural and biological characteristics*"[126]. A biometric characteristic requires mainly two attributes for an unambiguous recognition: *uniqueness* and *stability over time*. On the one hand side, a biometric trait shall be unique for every individual, i.e. the biometric trait must be different for different individuals. No two individuals shall share the same biometric characteristic. Only uniqueness of a trait allows to unambiguously recognise an individual. If a biometric trait is not unique, several individuals share the very same trait and biometric recognition will therefore be ambiguous. On the other side, a biometric characteristic shall be stable over time. Such stability allows a repeatable usage of the biometric characteristic for recognition. Only if stability is given, the biometric characteristic is usable for long term biometric recognition. If a biometric characteristic changes over time, biometric recognition needs to deal with the changes. For only slight changes in the biometric characteristic, biometric recognition may still be feasible. Significant changes over time may make biometric recognition impossible. If a trait changes fast, only short term usability of a biometric characteristic may be possible.

There are several biometric characteristics which fulfil both requirements. Each may be used for biometric recognition. However, only three of those characteristics are used most often in public responsibilities or in commercial domain: *fingerprint*, *face*, and *iris* [37].

Use cases for the application of biometric recognition are manifold. Recognition of individuals in public responsibilities happens for example during immigration across national borders. In the commercial domain biometric recognition has many use cases. Whenever an individual has to be recognized by another party, biometrics are applicable, e.g. when recognizing a bank customer at an ATM or when unlocking a smart phone.

One can distinguish mainly two scenarios of application of biometric recognition: *identification* and *verification*. The former scenario is defined as the "*process of searching against a biometric enrolment database to find and return the biometric reference identifier(s) attributable to a single individual*"[126]. The latter scenario is defined as the *"process of confirming a biometric claim through biometric comparison"*[126]. Thus, the difference between both is the interaction of the individual to be recognized. In case of verification the individual claims an identity, while in identification no identity claim needs to be made. This results in different workflows. In verification the biometric characteristic of the individual to be recognized is compared against the biometric characteristics attributed to the claimed identity. Therefore, only a single comparison is performed. In case of identification, the biometric characteristic of the individual to be recognized is compared against a set of biometric characteristics from known individuals. Two different modes can be distinguished for the scenario of identification: *closed set* and *open set* identification. In the former, the individual to be recognised is contained in the set of known individuals. Therefore, recognition of the individual is possible. In the latter, the individual does not have to be in the set of known individuals. Thus, recognition of the individual does not have to be possible.

### 1.2.2 Fingerprints as a Biometric Characteristic

Fingerprints fulfil both requirements to allow long term biometric recognition. Typically, the structure of the friction ridges is used as a *biometric feature* for recognition. Biometric features are actually defined as *"numbers or labels extracted from biometric samples and used for comparison"*[122]. The friction ridge structure alternates between *ridges* and *valleys*. The resulting structure is at least assumed to be unique for every finger and for every individual even though there is a discussion about the uniqueness [203]. In addition, for the aspect of stability "‘*the validity [...] has been established by empirical observations as well as based on the anatomy and morphogenesis of friction ridge skin*"’[203]. Of course, fingerprints alter over time. A finger grows during childhood and so does the fingerprint [96]. In addition, the skin of the finger ages. However, despite these changes the relevant information can be represented invariant to these changes. Significant changes of the relevant information will usually only happen, if the fingerprint was altered by severe accidents or damaged by intention to evade identification [323]. As long as the relevant information is not altered significantly, stability over time is therefore given.

Using fingerprints for biometric recognition has a long tradition. Systematic usage of fingerprints goes back far more than the 20$^{\text{th}}$ century [81]. Computer aided recognition of fingerprints has a tradition even as long as there is computer aided image processing. Fingerprint recognition is an active field of research with global attention. Over decades of sophisticated engineering, fingerprint recognition has evolved to a mature technology. Today, fingerprints can be processed fast and fingerprint identification systems can reach a high level of accuracy [310].

### 1.2.3 Fingerprint Identification System

The typical workflow in a fingerprint identification systems is as follows (see also Figure 1.11 for the schema of an entire system): The workflow starts with the acquisition of the fingerprint samples from the actual fingerprints. Typically, the fingerprint samples are digital images. There are several *modes* and *techniques* for the acquisition of a fingerprint sample.

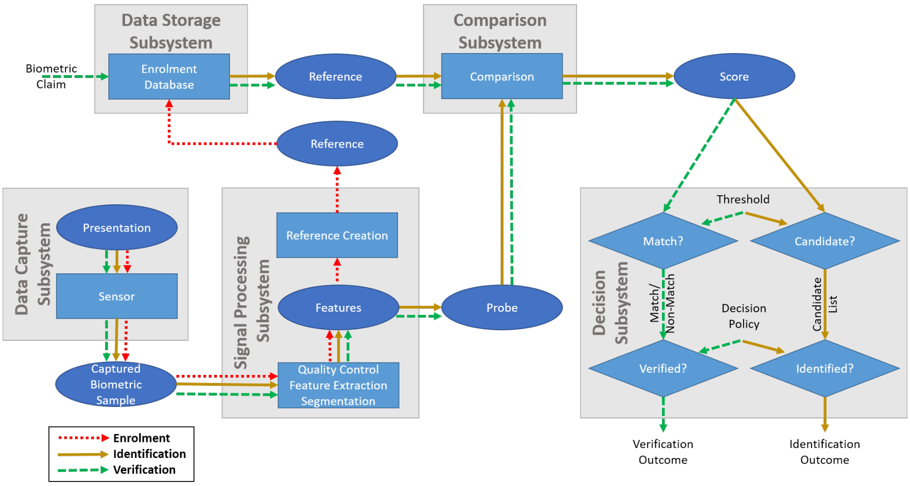


**Figure 1.11:** A generic biometric system contains subsystems for data capturing, signal processing, data storage, comparison, and decision. Different workflows apply for the modes enrolment, verification, and identification [120].

The acquisition technique describes how the fingerprint is acquired. There is a multitude of techniques. There are dedicated capturing devices, which are called livescanners. Such devices may have various technical details and may differ in the physical principles, which are used for acquisition, e.g. optical or capacitive sensors. Fingerprints may also be acquired with ink on paper. In some cases, the fingerprint is not even present at all: *latent fingerprints* or *fingermarks* are evidences from crime scenes (see Figure 1.12). In many cases, latent fingerprints contain only little information. The utility of such fingerprints for identification is only low. However, processing of latent fingerprints is a public demand, e.g. in recently published EU regulation 2018/1861 [231] They are acquired with forensic tools. Of course, any camera can be used for the acquisition of fingerprint, e.g. a smart phone camera.

The acquisition mode describes how a fingerprint is presented to the acquisition technique. The fingerprint may be placed *plain* on an acquisition surface. It also may be *rolled* over the surface of a capture device. In some cases, the fingerprint may be *swiped* over a line scanner, e.g. in smart phones and notebooks. While the former modes are all contact based, fingerprints may also be captured contactless.

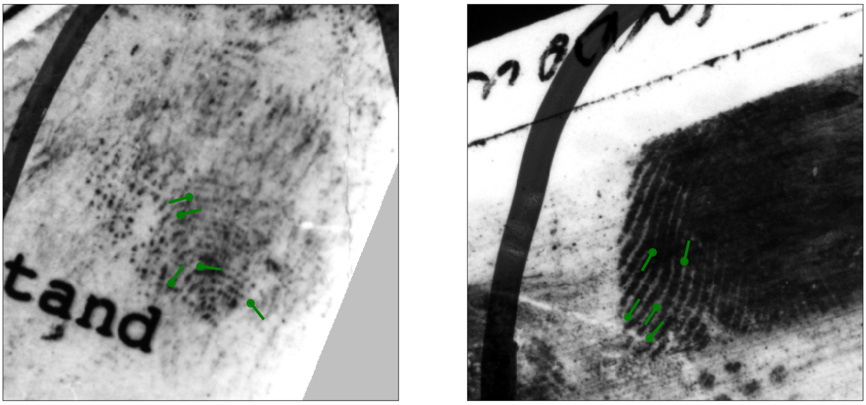


**Figure 1.12:** *Latent fingerprint* or *fingermarks* usually are of low quality and may contain only few minutiae (●–).

For latent fingerprints, the finger is not present during the acquisition process at all.

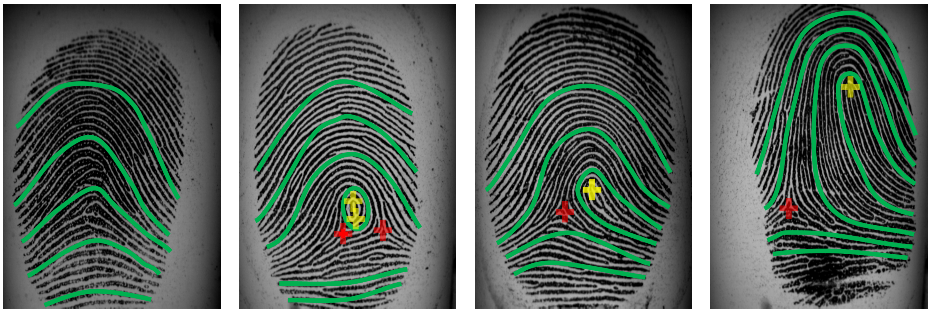
Commonly, *features* are *extracted* from the fingerprint, which are relevant for the process of the biometric *comparison*. What happens to these features depends on the use case. If the individual shall be *enrolled* to a fingerprint identification system, then the features will be stored as a *biometric data record* into a *biometric reference database*. If the use case is an identification, the features are compared in a biometric comparison against features from a biometric reference database of enrolled individuals. If the use case is a verification, the features are compared against those features enrolled to the biometric reference database for the claimed identity. Thus, in the latter two cases a biometric comparison will be carried out.

Each comparison generates a *biometric score*. This number is related to the probability of both two feature sets belonging to the same fingerprint. Based on this number a *candidate list* is generated in case of identification. In case of verification a *comparison decision* is made.

### 1.2.4 Features from Fingerprints

Fingerprints contain several features for biometric recognition. Usually, three levels of features are distinguished: $1^{st}$, $2^{nd}$, and $3^{rd}$ *level features*. These features represent a fingerprint from coarse to fine.

$1^{st}$ level features roughly describe a fingerprint at a global scale (see Figure 1.13). A typical example for such a feature is the orientation field. The orientation field is

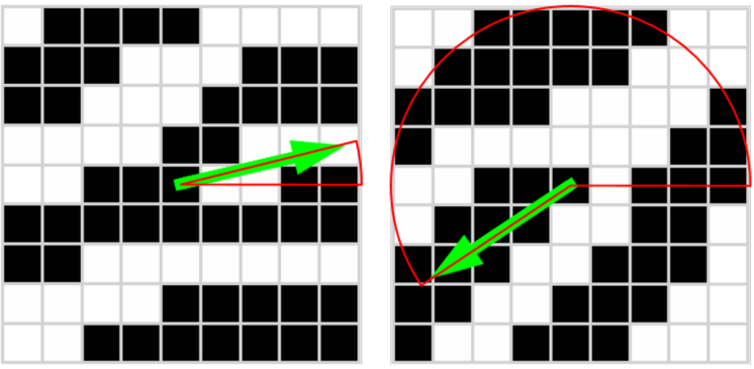


**(a)** Arch **(b)** Whorl **(c)** Right loop **(d)** Right loop

**Figure 1.13:** The orientation field of a fingerprint is a representation for the local orientations of the ridge structure. The presence or absence of singularities within the orientation fields significantly govern the orientations fields and builds typical patterns. Those singularities are *cores* +and *deltas* +. The green lines emphasize the flow of the ridges around those singularities. The relative positions of the singularities can vary the shape significantly within a pattern type (compare Figures 1.13(c) and 1.13(d))

a representation for the local orientation of the fingerprint ridges. Presence and position of *singularities* within the orientation field mainly govern the pattern of the orientation field itself. These typical patterns can be used as a feature in biometric recognition. Since this feature is not unique, it cannot be used for reliable recognitions. However, it allows exclusion during biometric comparisons, i.e. it allows to decide reliably, whether fingerprint samples are *non-mated*, i.e. "*paired biometric probe and biometric reference [...] are not from the same biometric characteristic of the same biometric data subject*"[126].

In contrast to 1st level features, 2nd level features are unique for every fingerprint. Thus, those features can be used for recognition. The classical features in fingerprint recognition are fingerprint *minutiae*. A minutia is a distinctive point in the fingerprint ridge structure. The two most prominent examples of minutiae are *ridge endings* and *ridge bifurcations* (see Figure 1.14). An ending is a point, where a ridge ends. A bifurcation is a point, where a single ridge splits up into two ridges. Rarely, *trifurcations* occur, i.e. where a single ridge splits up into three ridges. There are more known types of minutiae. All of them can be composed from two minutiae, e.g. an *island* or *point* is a very short ridge delimited by two ridge endings, which are very close together. Besides its type, a minutia has several further features. The two typical features are a minutia's *position* and its *direction*. The position represents *where* the minutia is located within a fingerprint. The direction can be derived from the minutia's type and the local orientation of the



**(a)** Ridge Bifurcation

**(b)** Ridge Ending

**Figure 1.14:** Minutiae are characteristic points in the fingerprint ridge structure. The points where single ridges split up into two ridges are called a *bifurcations* (1.14(a)). An *ending* is the point, in which a ridge ends (1.14(b)).

fingerprint ridges at the position of the minutia. A single minutia does not make a fingerprint unique. The relations of positions, directions, and types of all minutiae in a fingerprint make it unique.

While 2$^{\text{nd}}$ level features are already quite small and local structures, 3$^{\text{rd}}$ level features are even more detailed features of a fingerprint. The typical 3$^{\text{rd}}$ level feature is a *sweat pore*. Those pores are located on the ridges. High resolutions ($> 1000\,\mathrm{dpi}$) allow to capture this tiny detail reliably. Only for some few sensors with high utility properties pores can be identified even in $500\,\mathrm{dpi}$ samples. Since the resolution required for a reliable acquisition of pores usually exceeds the standard resolution of $500\,\mathrm{dpi}$, pores are not always detectable and are therefore only rarely used for recognition. However, pores are very distinctive for a fingerprint. Like minutiae, the positions and shapes of pores are unique for every fingerprint.

There is a classical workflow, which allows to extract the fingerprint minutiae from a fingerprint sample. It consists of several processing steps. The workflow can roughly be sketched as follows. First, the orientation field is extracted. Second, the local distance between the ridges is estimated based on the orientation field. Therefore the distance between adjacent ridges is measured orthogonal to the local orientations. Third, an enhanced instance of the original fingerprint sample is generated. To do so, a filtering is applied which takes into account the local orientations and the local frequencies, which can be derived from the local distances between ridges. Usually, *Gabor* filters are used for the enhancement as they are able to take orientation and frequency into account [110]. The resulting enhanced image contains the clear structure of the fingerprint ridges. Forth, the structure is reduced to a graph-like structure by a skeletonization [128]. Finally, the minutiae are extracted directly from the graph. Information on the position and the type of a minutia can be extracted from the graph. The combination of the local structure of the graph and the orientation field allows to extract information on the direction of a minutia [121]. Figure 1.15 visualizes this workflow of extracting minutiae from fingerprint samples.

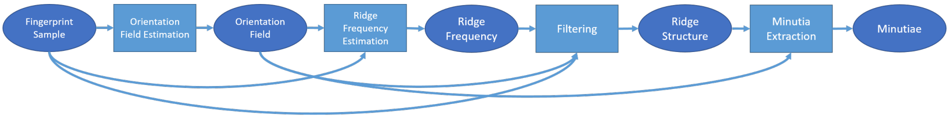


**Figure 1.15:** The classical approach to extract minutiae from a fingerprint consists of several processes. First, the orientation field is estimated. Then, local ridge frequencies are estimated based on information from the fingerprint sample and the orientation field. In the next step, the sample can be filtered taking local orientation and frequencies into account. Finally, minutiae can be extracted from the resulting ridge structure. The orientation of each minutiae can be derived from the orientation field.

This workflow only hinted here is straight forward and easy to implement. However, there are several challenges in the extraction of fingerprint minutiae.

Many processes like the estimation of the local frequencies rely on an accurate estimation of the orientation field. Thus, the estimation of the orientation field is a crucial task in this classical workflow. Estimations in areas of high curvature are more difficult, because the variation is higher there.

For fingerprint samples of good quality the extraction is not challenging at all. Even simple gradient based approaches work well for good quality images (see Table 8.1). However, in case of low quality fingerprint samples, the extraction will be more challenging. Low quality fingerprint samples may have several reasons. Skin diseases, environmental conditions disturbing the acquisition process, or simply inappropriate acquisition equipment are only some aspects, which might result in low quality fingerprint samples. Estimation of the orientation field or estimation of the local frequency in areas of low image quality may fail. Filtering with erroneous information on orientation or frequencies will result in an erroneous enhanced ridge structure. The final graph-like structure may be disturbed significantly in this case. The disturbance can affect all aspects of information of a minutia. For example, the direction and position of the minutia can be inaccurate. Beside these inaccuracies, another type of erroneous extraction may even be more severe: unreliability. Spurious minutiae may be extracted at locations where there is no minutia at all (see Figure 1.16). On the other hand side, actual minutiae might be missed by the feature extraction.

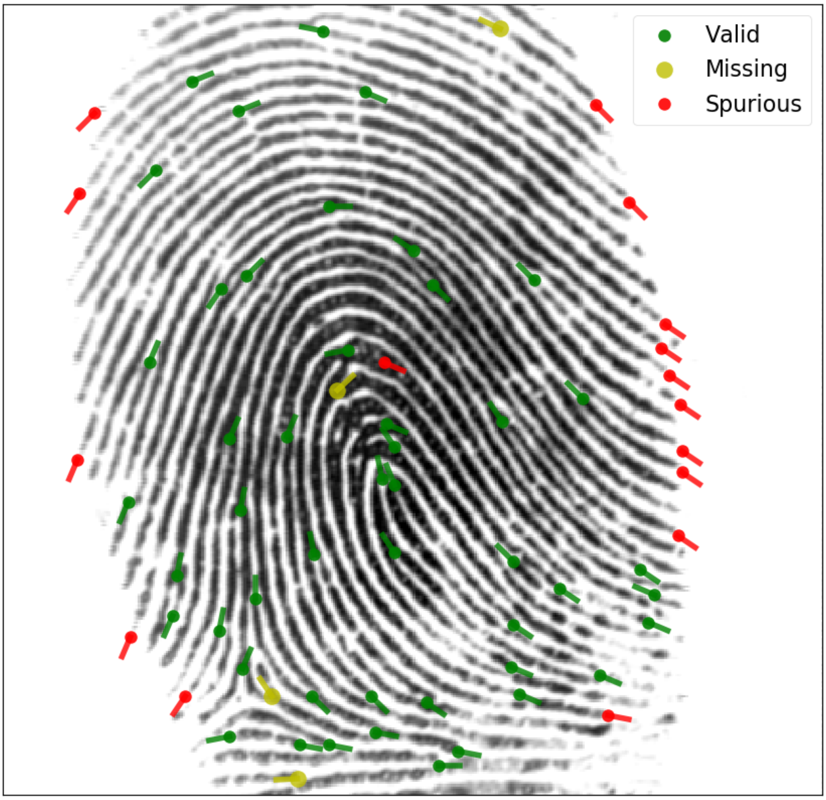


**Figure 1.16:** Feature extractor MINDTCT finds many valid minutiae close to actual minutiae (●-). However, minutiae are not always extracted semantically conform to actual minutiae: Some actual minutiae are missing (●-) and also some spurious minutiae (●-) were introduced. Ground truth data was manually marked-up [148].

An inevitable aspect is the fact, that a fingerprint sample is only a sample. In general, not all information of the fingerprint is included in the sample. Thus, the extracted features will be incomplete. Incompleteness will be severe, if only a small part of the fingerprint is present in the sample. However, this challenge of incompleteness has to be dealt during the biometric comparison process.

Figure 1.17 visualizes the entire workflow of transforming a fingerprint into biometric features.

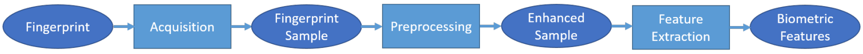


**Figure 1.17:** Extracting features from fingerprints represents the first part of the classical machine learning workflow (compare to Figure 1.1). The workflow consists of acquisition, preprocessing, and the actual feature extraction.

### 1.2.5 Fingerprint Comparison

Classical comparison of fingerprints uses minutiae as main features. Each fingerprint sample is represented by a set of minutiae, which were extracted from it. Each set of minutiae can be interpreted as a point cloud. Thus, fingerprint comparison can be seen as a point cloud similarity estimation problem.

As these point clouds are usually unaligned, one has to determine the relative position and orientation of the point clouds to each other. An exhaustive comparison of the entire point clouds is too expensive in terms of computational effort. In the next step, the corresponding neighbourhoods are inspected for consistency with respect to their local relation. After some corresponding minutiae have been determined, the point clouds can be aligned, such that both point clouds overlap best. Finally, the two point clouds can be inspected for their similarity.

There are also challenges in the biometric comparisons. Some are effects of the challenges of biometric feature extraction. Others are intrinsic to the comparison process.

The challenges inherited from feature extraction are manifold. Due to the fact, that a fingerprint sample is only a digital representation of the fingerprint, biometric comparison has to deal with the differences between samples and actual fingerprints. For example, the acquisition process might have introduced distortions into the fingerprint sample. Distortions will occur for example, if the finger is pressed on an contact-based capture devices. For *finger photos* from contact-less capture devices, no distortions occur. However, in this case biometric comparison has to deal with the degree of freedom arising from the free movement of the finger during capturing. Eventually, all challenges can be summarized to the fact, that the point clouds are unreliable, inaccurate, and incomplete.

The intrinsic challenges of the comparison process can be described best, if one inspects and understands the outcome of a biometric comparison. The outcome of a single biometric comparison of two fingerprint samples is a biometric score. This score is more or less related to two different probabilities. First, it is related to the probability, that the two compared fingerprint sample are *mated*, i.e. they belong to the very same fingerprint. Second, this score is also related to the probability, that the two compared fingerprint samples are non-mated, i.e. they belong to two different fingers.

The score is the base for a decision. For example, in the case of a verification attempt the claimed identity might be accepted, if the score exceeds a specific threshold. In this case the threshold can be interpreted as a required level of confidence, that the two compared fingerprint samples are mated or vice versa that they are not non-mated.

## 1.3 Evaluation Methodology

The decisions whether two fingerprints are mated are not always correct. There are two types of errors: *false mates* and *false non-mates*. If two fingerprints are actually mated but decided by the decision subsystem to be non-mated, these will be false non-mates. Vice versa if two fingerprint samples are non-mated but decided to be mated, those will be false mates. The central challenge in fingerprint comparison is of course to cause as little decision errors as possible. However, there are three aspects which are most important to describe a *Automated Biometric Identification System* (ABIS): accuracy, accessibility, and speed [119].

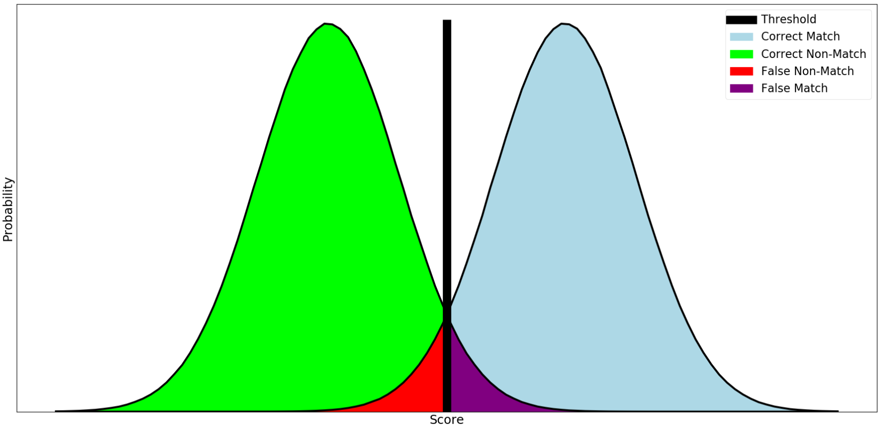
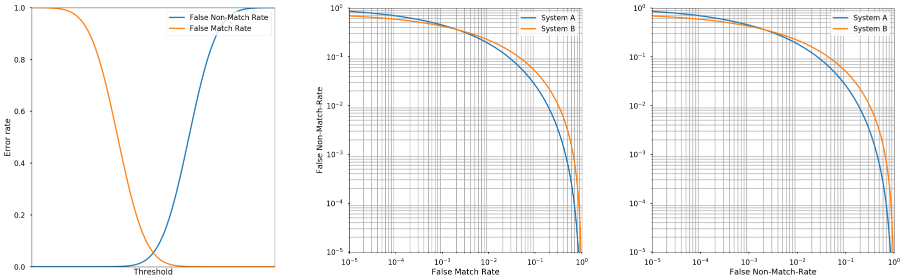


**Figure 1.18:** Biometric decisions are made based on the biometric scores and a threshold to be applied. If the score of a non-mated comparison exceeds a given threshold, a false match error will occur. If the biometric score of mated comparison does not exceed a given threshold, this will result in a false non-match error.

The aspect of accuracy is usually assessed in terms of error rates, i.e. the expected probabilities for the occurrence of errors. There are two error rates, which can be derived for the types of errors being made during the biometric comparison. The probability for an erroneous decision for a comparison being non-mated is represented by the *false non-match rate* (FNMR). The respective error rate for comparisons being erroneously decided to be mated is the *false match rate* (FMR). FNMR and FMR consider only errors resulting from the biometric comparison and therefore are used for *algorithm evaluation*. When an entire ABIS is analysed, acquisition error need to be taken into account, too. In this case *false acceptance rate* (FAR) and *false rejection rate* (FRR) are used for system evaluation. Neither FNMR nor FMR take prior probabilities for mated or non-mated comparisons into account. The decision whether two samples match is obtained by thresholding the comparison against a predefined score value. Thus, both FNMR and FMR are in general depending on a threshold. Throughout this thesis a technology evaluation according to ISO/IEC 19795-1[118] is conducted, where data-subject to capture device interaction and corresponding failure to capture cases are not investigated. Thus all evaluations report algorithm performance. However, the relevant metric for the former type of failures is the *Failure-To-Enrol Rate* (FTE) which represents the *"proportion of the population for whom the system fails to complete the enrolment process"* [118]. The latter type of failures is tracked by the *Failure-To-Acquire rate* (FTA), which stands for the *"proportion of verification or identification attempts for which the system fails to capture or locate an image or signal of sufficient quality"* [118]. Figure 1.18 visualizes two exemplary distributions of comparison scores and the respective errors to be made for both types of comparisons. FNMR and FMR can be plotted versus the threshold to visualize a system's error rate when the threshold is varied (see Figure 1.19(a)). As error rates are usually quite low, they are plotted in a logarithmic scale (see Figure 1.19(b)). Such plots allow an inspection of the behaviour of a single system with respect to the threshold. In general, different systems vary in their behaviour with respect to the threshold. This complicates comparison of systems. *Detection error tradeoff* (DET) curves are independent of any threshold and allow comparison between different systems (see Figure 1.19(c)) [118]. FNMR, FMR, and DET curves are used for the scenario of verification.



**(a)** Linear **(b)** Logarithmic **(c)** DET

**Figure 1.19:** Based on a variable threshold, the expected error rates will vary (1.19(a)). As the error rates can be relative small, a logarithmic scaling may be reasonable (1.19(b)). Detection error tradeoff (DET) curves allow an comparison of different systems (1.19(c)). These plots are common tools to inspect biometric systems in case of a verification scenario.

For the scenario of identification, other metrics are applied. Usually, candidate lists of potentially mated individuals are generated.[2] Thus, the order of the comparison scores is usually more important than their actual values. For such cases, a rank-based assessment is reasonable. Typically, *cumulative match characteristic* (CMC) curves are used for visualization of rank-based analysis (see Figure 1.20). In this case, two rates are mainly of interest: the *False-Negative Identification-Error Rate* (FNIR) and the *False-Positive Identification-Error Rate* (FPIR) . For given rates FMR, FNMR, and FTA, both rates FNIR and FPIR can be approximated as follows [118]:

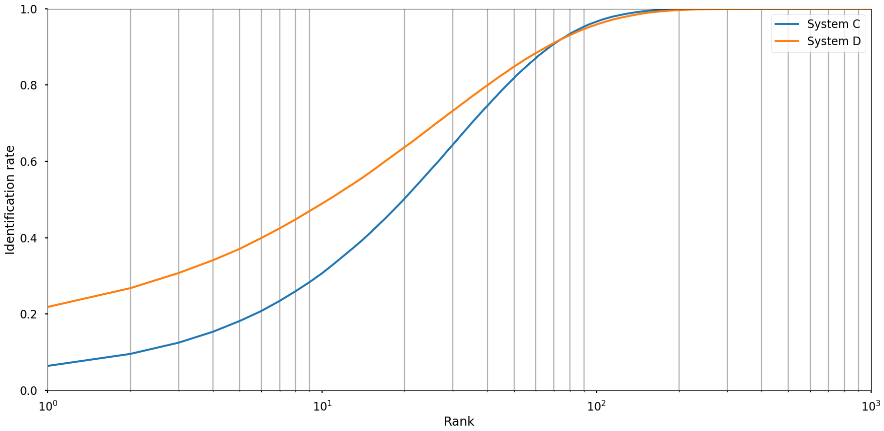


**Figure 1.20:** *Cumulative match characteristic* (CMC) curves allow a rank-based comparison of different systems in the identification scenario. CMCs consider the rank of biometric scores. The x-axis may also be displayed relative to the size of the reference database to be searched.

$$\text{FNIR} = \text{FTA} + (1 - \text{FTA}) \cdot \text{FNMR} \tag{1.7}$$

$$\text{FPIR} = (1 - \text{FTA}) \cdot (1 - (1 - \text{FNMR})^N) \tag{1.8}$$

where N is the size of the biometric reference database. If one assumes FTA $= 0$ to focus on the algorithm errors, the equations simplify to the following:

$$\text{FNIR} = \text{FNMR} \tag{1.9}$$

$$\text{FPIR} = 1 - (1 - \text{FNMR})^N \tag{1.10}$$

The rate FPIR therefore significantly depends on the size of the biometric reference database $N$.

[2] Actually, thresholds are also considered in some generation policies for candidate lists.

In some cases, *pre-selection algorithms* are used to reduce the workload for the actual identification algorithm. Such an algorithm is defined as an *"algorithm to reduce the number of templates that need to be matched in an identification search of the enrolment database"* [118]. For pre-selection algorithms, there two rates central metrics: the *penetration rate* (PR) and the *Pre-Selection Error Rate* (PSER) . The former is defines as the "measure of the average number of pre-selected templates as a fraction of the total number of templates" [118]. The later is the occurrence rate of *pre-selection errors*. A single pre-selection error is defined as the "error that occurs when the corresponding enrolment template is not in the preselected subset of candidates when a sample from the same biometric characteristic on the same user is given" [118].

Defining a threshold or the policy for candidate list generation for a biometric system influences a systems behaviour significantly. There is a typical tradeoff between convenience and security in an ABIS. For more secure verification systems, one will likely allow a higher FNMR to enforce a low FMR. Convenient verification systems will vice versa allow a higher FMR to ensure a low FNMR [118].

For the case of pre-selection algorithms there are related metrics, which are used for the evaluation at benchmark framework FVC-ongoing. At this benchmark, two metrics are central: PR and *Error Rate*.[3] The benchmark interprets PR as *"the portion of database that the system has to search on the average (corresponding to the average length of the candidate lists)"*[3] and defines ER *"as the percentage of searched fingerprints that are not found"*[3]. The benchmark evaluates candidate lists returned for each identification search. The relation of ER and PR depends on an additional parameter $\text{Max}_{\text{PR}}$. This parameter controls the maximum percentage of the candidate list length relative to the size of the entire biometric reference database $N_{\text{DB}}$. Let $N_q$ be the number of searches and let $C_i$ be an ordered list of $N_i$ candidates returned for the i-th search. Then the effective length $L(i, \text{Max}_{\text{PR}})$ of a candidate for i-th search containing at most $\text{Max}_{\text{PR}}$ items is defined as follows:

$$L(i, \text{Max}_{\text{PR}}) = \min(\text{Max}_{\text{PR}}, N_i) \tag{1.11}$$

while a single error $\text{Err}(i, L)$ for such a list is defined as:

$$\text{Err}(i, L) = \begin{cases} 1 & \text{if the i-th mate is not in the first L candidate of } C_i \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

[3] https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/BenchmarkAreas/BenchmarkAreaFIDX.aspx

For a given maximum percentage $\text{Max}_{\text{PR}}$, the rates ER and PR are then evaluated as follows:

$$\text{PR}(\text{Max}_{\text{PR}}) = \underbrace{\frac{1}{N_q}\sum_{i=1}^{N_q}\frac{L(i,\text{Max}_{\text{PR}})}{N_{\text{DB}}}}_{\text{Average PR considering at most Max}_{\text{PR}}\text{ candidates}} \tag{1.13}$$

$$\text{ER}(\text{Max}_{\text{PR}}) = \underbrace{\frac{1}{N_q}\sum_{i=1}^{N_q}\text{Err}(i,L(i,\text{Max}_{\text{PR}}))}_{\text{Average ER considering at most Max}_{\text{PR}}\text{ candidates}} \tag{1.14}$$

FVC-ongoing finally reports two rates: $\text{ER}_{100}$ and $\text{ER}_{1000}$. The former represents the lowest PR for $ER \leq 10^{-2}$ and the later represents the lowest PR for $ER \leq 10^{-3}$. In addition, the *Incremental Search* scenario is defined as the scenario *"where an ideal matching algorithm is used to stop the search as soon as the right candidate is retrieved, is considered. In such a scenario there are no retrieval errors since, in the worst case, the search can be extended to the whole database, and the average penetration rate is reported as"*[3] the rate IS.

The benchmark FVC-ongoing is essential for the evaluation of pre-selection algorithms. Therefore, this work focusses on the metric used in this benchmark. The rate IS is considered to be most important, since it indirectly aims for an optimal workload reduction while allowing no false non-identification errors in the closed set identification scenario.

The aspect of accessibility monitors the limitations of an ABIS due to the individuals themselves. There may be individuals who are not able to provide sample, e.g. if they are actually missing some fingers.

Of course, the aspect of transaction times is important, too. If an ABIS cannot provide recognition within a reasonable and appropriate time, it might be useless. This might be challenging for example in case of the identification scenario and large databases. Identification might simply take too much time. But it also may be challenging in the verification scenario, if the computational resources are sparse, e.g. on smart cards.

## 1.4 Challenges

There are more challenges in fingerprint recognition than those challenges mentioned above for biometric feature extraction and biometric comparison.

Nowadays, ABIS get larger and larger.[4] Each and every new enrolled individual enlarges the biometric enrolment database. The enlargement increases the work load of the ABIS for every identification attempt. This in turn degrades the aspects of the performance of systems described earlier: The more individuals are enrolled, the more time a query in the biometric enrolment database will take. Thus, the response time will increase and speed decreases. The more individuals are enrolled, the more likely it is for a ABIS to make errors, too: Especially the FPIR decreases significantly in the size of the dataset (see Equation 1.10). A famous example for a false positive identification due to large databases is related to the Madrid train bombing in March 2004. Since the bomber could not be found within Europe, the search was widened globally. Due to this worldwide search, the size of the biometric reference database of possible candidate was tremendous. A candidate was found, whose fingerprint had a high likelihood to a trace found at the crime scene. Despite this high likelihood, which was also assured by forensic experts, he was not the bomber and was arrested wrongly [252]. It was a false-positive identification, which might be inevitable for large biometric reference databases, when other means of identification are not taken into account. Utilized techniques need to keep this scaling issue to a reasonable order.

The multitude of variable components is an additional source of challenges. *Interoperability* is challenging in many ways. There are standardized formats for interchange of fingerprint minutiae data [121]. Furthermore, there are three levels of conformance tests for these formats. While conformance testing level 1 checks the validity the structure of the format, level 2 deals with the internal consistency of the stored data. However, different acquisition techniques result in varying characteristics of the fingerprint samples [244]. Conformance testing level 3 actually deals with variance, since this level is about semantic conformance of the stored data to the actual fingerprint [36][4]. Feature extraction or preprocessing of fingerprint samples must be able to deal with such differences. Biometric features from different acquisition modes can be very dissimilar. A biometric comparison of fingerprint samples acquired as plain fingerprints needs to be at least slightly different from a comparison dealing with fingerprint samples acquired as rolled fingerprints. Different feature extractors extract biometric features differently [98]. The differ-

[4] The Aadhaar project has today's large scale ABIS par excellence. It aims to enrol the entire population in the Republic of India [84].

ences are systematic and can be significant. Biometric comparison of biometric features with such varying characteristics is challenging.

Low quality of fingerprint samples is a central challenge in fingerprint recognition. The lower the quality of a fingerprint sample is, the lower is its biometric utility, i.e. its suitability for a reliable biometric recognition [123]. There are several causes for low quality, e.g. unconstrained acquisition conditions. Low quality may also be the result of skin issues due to work or skin diseases. Latent fingerprint samples from evidence at crime scene are examples for very low quality. Biometric sample quality for fingerprint images can be estimated by using ISO/IEC standard 29794-4 [124]

Beside the described challenges there are more open issues, of course. For instance *Presentation Attack Detection* [125] and *Biometric Template Protection*[5] [129] will be essential for long term credibility and applicability in high security ABIS.

[5] Once a biometric template is "stolen", it might be used to attack a ABIS. An individual's fingerprint, face, or iris is not changeable. Instead of storing plain biometric data, combinations of biometric data with revocable helper data are stored in biometric data records. This allows regeneration of a biometric data record.

# Chapter 2

# Contribution

## 2.1 Research Objectives

Classical fingerprint recognition consists mainly of the two processes: biometric feature extraction and biometric comparison. Both processes are based on best practices in engineering of image processing and pattern recognition. DL achieved remarkable improvements in the domains of image processing and pattern recognition. Despite these successes, DL had only very little application in fingerprint recognition until recently. Thus, the main objective was the comparison of handcrafted approaches to DL approaches in fingerprint recognition:

*"Comparison of hand-crafting and Deep Learning approaches in fingerprint recognition"*

However, this core objective is very general. Replacing the entire process of fingerprint recognition by a new one would be way too difficult to evaluate. Evaluation of all aspects concerned would not be feasible. However, biometric feature extraction as well as biometric comparison are no monolithic processes. Both assemble from *modules* performing single processing steps, e.g. estimation of the orientation field. In conventional biometric information systems, each module is based on engineered features. Thus, each processing step is a candidate for being replaced by a process, which makes use of DL. Evaluation on the level of single modules is reasonable and feasible.[1]

[1] The influence of a single processing step on the entire workflow of fingerprint recognition cannot be identified explicitly. However, the performance of the entire biometric recognition is the most central fact for application. Therefore, the performance shall always be kept in mind.

Since the objective is very general, several more specific research objectives can be deducted from the challenges, which have been identified for fingerprint recognition (see Section 1.4). In particular, research objectives arise with respect to dealing with fingerprint samples of *low quality*, to estimation of the *orientation field*, and to application of DL for *efficient processing structures* for fingerprint recognition systems.

## 2.2 Research Questions

The core objective just spans the larger frame of the research. The actual research questions address the more specific research objectives in Section 2.1. The three following research questions were derived:

$\mathcal{RQ}_1$ *"Can DL outperform classical fingerprint sample enhancements?"*

$\mathcal{RQ}_2$ *"Can DL be used for a better orientation field estimation?"*

$\mathcal{RQ}_3$ *"Can DL provide solutions for efficient processing structures?"*

## 2.3 Publications

These publications addressing the research questions are part of this dissertation:

$\mathcal{RQ}_1$ *"Can DL outperform classical fingerprint sample enhancements?"*

[265] P. Schuch, S. Schulz, and C. Busch. Survey on the impact of fingerprint image enhancement. *IET Biometrics*, 7:102–115(13), March 2018. ISSN 2047-4938. URL `http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2016.0088`

[257] P. Schuch, S. Schulz, and C. Busch. De-convolutional auto-encoder for enhancement of fingerprint samples. In *6th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–7. IEEE, 2016

[258] P. Schuch, S. Schulz, and C. Busch. Minutia-based enhancement of fingerprint samples. In *International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2017

$\mathcal{RQ}_2$ *"Can DL be used for a better orientation field estimation?"*

[259] P. Schuch, S.-D. Schulz, and C. Busch. Convnet regression for fingerprint orientations. In *Scandinavian Conference on Image Analysis*, pages 325–336. Springer, 2017

[260] P. Schuch, S.-D. Schulz, and C. Busch. Deep expectation for estimation of fingerprint orientation fields. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 185–190. IEEE, 2017

[261] P. Schuch, S.-D. Schulz, and C. Busch. Intrinsic limitations of fingerprint orientation estimation. *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2017

$\mathcal{RQ}_3$ *"Can DL provide solutions for efficient processing structures?"*

[256] P. Schuch. Survey on features for fingerprint indexing. *IET Biometrics*, June 2018. ISSN 2047-4938. URL `http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2017.0279`

[264] P. Schuch, J. M. May, and C. Busch. Unsupervised learning of fingerprint rotations. In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553096

[262] P. Schuch, J. M. May, and C. Busch. Estimating the data origin of fingerprint samples. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553235

[263] P. Schuch, J. M. May, and C. Busch. Learning neighbourhoods for fingerprint indexing. In *2018 14th Signal Image Technology and Internet-based Systems (SITIS)*, pages 1–6, Nov 2018

## 2.4 Structure of the Dissertation

The rest of this dissertation is structured as follows: The next three parts deal with one research question each. Part II deals with fingerprint sample enhancement. Fingerprint orientation field estimation is the topic of Part III. Part IV is about the application of DL for efficient processing structures in fingerprint recognition systems. This part IV deals especially with Fingerprint Indexing. Each of these three parts starts with a section regarding its motivation. Then, a section is dedicated to provide an overview on state of the art in the respective topic. The next section describes this work's contribution to the respective topic, which exceeds the state of the art. Then, all relevant peer-reviewed publications follow. Each part will be concluded by its own section, which summarizes the respective topic of the part. Finally, Part V concludes the entire work.

# Part II

# Fingerprint Sample Enhancement

# Chapter 3

# Introduction

## 3.1 Motivation

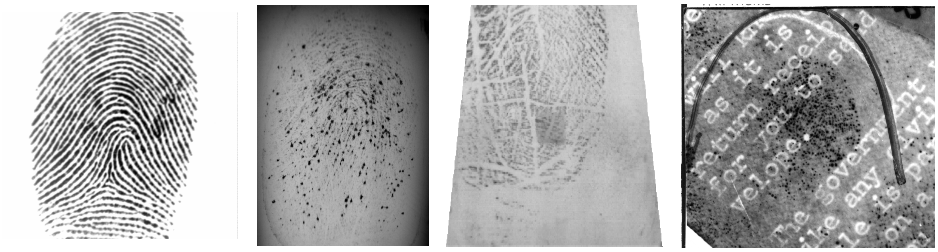
Insufficient quality of fingerprint samples is a critical issue in fingerprint recognition [279]. Low quality may have several reasons: issues during the *capture process*, *skin issues*, or *insufficient cooperativeness* of the individual.

During the capture process several aspects may result in low quality samples. Environmental influences can disturb the acquisition process. For example, bright ambient light can disturb the capture process when using optical live-scanners. This may in turn result in over-exposure or low contrast of the fingerprint samples. While new fingerprint scanners may produce high quality fingerprint samples, it is likely that the quality of the capture device itself may decay over time. However, the influence of ageing of fingerprint sensors is assumed to be irrelevant [145]. Contamination of acquisition surfaces or fingerprints with dirt can also result in low quality fingerprint samples [203].

Another challenge during the capture process may be distortions of the fingerprint structure. For example, such distortions may be intrinsic to the acquisition process. Any contact-based capture process distorts the fingerprint structure automatically. Such distortions are usually larger when the fingerprint needs to be moved during the acquisition, e.g. when the fingerprint is rolled over a surface. But this may also arise from inappropriate ergonomics of the capture device [165]. The device might for example be out of place, such that it cannot be reached without any difficulty. In such cases, the fingerprint may be severely distorted. Distortions are usually present all the time. Even though such distortions are usually no challenge for the biometric feature extraction, they might be very challenging for the biometric comparison.

*Latent fingerprints* or *fingermarks* are usually also of very low quality (see Figure 3.1(d)). This is due to the acquisition process, in which the fingerprints are lifted from evidences. In some cases only very small parts of the fingerprints are visible at all. Latent fingerprints can have a very low signal to noise ratio, i.e. the background and other disturbing noise may be more present than the relevant fingerprint ridge structure. Processing such latent fingerprints of low quality is most likely one of the hardest challenges in fingerprint recognition.

Skin issues can also be a challenge to the acquisition of fingerprints (see Figure 3.1(c)) [235]. Some skin diseases may results in dry or worn off skin. Acquisition of dry skin is usually more challenging, since the contact between acquisition surface and fingerprint is reduced. Worn off skin usually results in a less clearer presence of the relevant fingerprint ridge structure. For some diseases the outer skin tissue, the epidermis, may be severely damaged. Acquisition of such fingerprints may even be almost impossible. Besides skin diseases, manual labour as in agriculture or in the construction industry can also make the skin dry and worn off. Dry and worn off skin can also simply arise from the ageing of the skin. The biometric quality of fingerprints in general tends to degrade over time [24]. Older people tend to have low quality fingerprints. Low quality due to skin issues will therefore be present in a biometric identification system in every case, if a system shall be inclusive and not rejecting anyone due to skin issues.



**(a)** Good quality **(b)** Dry Skin **(c)** Skin Issues **(d)** Latent fingerprint

**Figure 3.1:** The quality of fingerprint samples varies significantly. While the fingerprint ridge structure is clearly perceivable in high quality images (3.1(a)), samples of low quality are challenging to biometric feature extraction. Low quality may arise for various reasons, e.g. due to dry skin (3.1(b)) or skin issues (3.1(c)). Latent fingerprints may be hardly perceivable due to significant background noise (3.1(d)).

Last but not least, an individual simply might not want its fingerprints to be acquired. This may happen for example, if a criminal wants to evade identification. In such cases, the individual may intentionally distort or contaminate its fingerprints to reduce the biometric sample quality.

There are tools which allow to assess the *biometric quality* of fingerprint samples. A biometric quality is a metric, which allows to estimate the *biometric utility* of a sample. The quality is defined as the *"degree to which a biometric sample fulfils specified requirements for a targeted application"* [123]. It can be interpreted as a sample's suitability for usage in a biometric recognition process. A high value of biometric quality shall indicate a high probability, that a sample can be used for a reliable and accurate biometric recognition process. Usually, all commercial biometric feature extraction algorithms provide their own methods for the assessment of a sample's biometric quality. Unfortunately, inter-operability between metrics of different commercial vendors must be assumed limited. But there are publicly available tools for assessment of the biometric quality. The most prominent tool is the *NIST Fingerprint Quality* (NFIQ), which comes in two generations: NFIQ1[280] and NFIQ2.0[222][223], where the latter is the de facto standard worldwide. NFIQ2.0 is a development, which overcomes limitations of its predecessor, e.g. by providing a finer granularity in the range of quality scores. Both metrics are de facto standard and are widely used in deployed systems all around the world [18]. Those tools allow to apply quality control on the fingerprints, which are to be processed. However, quality control cannot be applied in all scenarios. In some legacy systems quality control may not have been applied at all.

The impact of low quality fingerprint samples on a biometric recognition system can be severe. Like always in computer science, also for fingerprint recognition holds: *garbage in, garbage out*. In the worst case, features extracted from low quality fingerprint samples may be completely useless for biometric recognition. Low quality samples result more likely in incorrect biometric decisions than high quality samples. If a fingerprint sample of low quality is enrolled to a biometric identification system, it is usually harder to be found than a high quality fingerprint sample. Thus, low quality samples are more likely to result in false non-matches.

However, counter-measures can be taken to improve the biometric quality of a fingerprint sample. The fingerprint samples acquired from a fingerprint can be preprocessed before feature extraction is carried out. Figure 3.2 highlights this process in the feature extraction workflow. As a preprocessing is usually applied to enhance the biometric quality of a fingerprint sample, such a process may therefore also be called a *fingerprint sample enhancement*.

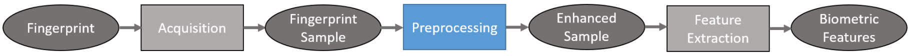


**Figure 3.2:** Preprocessing of fingerprint samples can be part of the workflow. Preprocessing can be used for an enhancement of a sample with respect to its biometric quality. An enhanced sample may have a higher biometric utility.

There are several ways to enhance a fingerprint sample. A typical enhancement approach is to enhance the relevant fingerprint ridge structure. The feature extraction process will extract features the more reliable and the more accurate, the clearer the fingerprint structure is. Another common preprocessing is a foreground detection. A foreground detection can allow a masking of the relevant fingerprint while suppressing the irrelevant background. This can also improve the performance of the biometric feature extraction process, since this process will not be distracted by any structures in the background. Finally, it is possible to apply a dedicated process to deal with systematic, known challenges of given fingerprint samples. For example, fingerprint samples on paper often contain lines and writings on the paper form, which was used for the acquisition. Elimination of such systematic noise will increase the biometric quality. However, low biometric quality due to distortions cannot be compensated by fingerprint sample enhancement. Thus, fingerprint comparison needs to deal with such issues.

## 3.2 State of the Art

### 3.2.1 How to assess fingerprint sample enhancement?

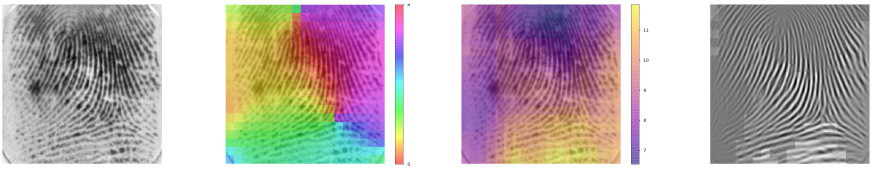
A fingerprint sample enhancement shall improve a sample's biometric utility. There are biometric quality metrics like NFIQ1 and NFIQ2.0 available. The applicability of such metrics to enhanced fingerprint samples is possible at least to some degree [265]. However, there are no independent benchmarks available to assess the impact of an enhancement on biometric utility.

A typical workaround is to evaluate the biometric performance explicitly. The impact of enhancement can then be assessed by comparison of biometric performances when using original samples and when using enhanced samples. Thus, biometric error rates are considered as an appropriate means to assess fingerprint sample enhancement. The lower the quality of a fingerprint, the more important is an enhancement. Thus, latent fingerprints are often used for assessment, since those are considered most challenging [42][276][282]. However, fingerprint sample enhancement can be important for good quality samples as well.

### 3.2.2 Hand-crafted approaches

Sophisticated engineering yielded an almost de facto standard method of fingerprint enhancement called *Contextual Filtering* [110]. This kind of enhancement makes strongly use of domain knowledge about the structure of fingerprints. Such domain specific knowledge allows to apply a dedicated enhancement to fingerprint samples.

Contextual Filtering basically consists of three central, consecutive processes. First, the *orientation field* is estimated. The orientation field is a representation for the local orientation of the fingerprint ridge structure (see Sections 1.2.4 and 8.1). If the orientation field is known, the flow of the ridge structure is known. In a second step, the distances between adjacent fingerprint ridges are estimated. For this estimation, knowledge about the orientation field is used: The distance is estimated by assessing the ridge structure perpendicular to the local orientation. As the fingerprint ridge structure is assumed to be similar to a cosine wave, cross sections perpendicular to the local orientation reveals a cosine-like structure. This allows to estimate the local distances between the fingerprint ridges. The distance can also be represented in terms of *local frequencies*. Now that the orientation field and local frequencies are known, a specialized filtering can be applied as a final enhancement step. Usually, *Gabor filters* are used for the filtering. Such filters take into account frequencies and orientations and are therefore well suited for an enhancement. The fingerprint ridge structure in the resulting enhanced fingerprint sample is usually clearly visible. However, the lower a fingerprint sample's quality is, the more challenging is an accurate enhancement.



**(a)** Original **(b)** Orientation Field **(c)** Frequencies **(d)** Enhanced

**Figure 3.3:** Contextual Filtering is a widely used technique for fingerprint sample enhancement. It consists of three steps: First, the orientation field is estimated 3.3(b). Then local frequencies are estimated 3.3(c). Finally, knowledge of local orientations and frequencies allows to apply appropriate filters. The enhanced sample 3.3(d) has a clearer ridge structure.

Contextual Filtering is likely to be "*the most widely used technique for fingerprint image enhancement*"[203]. It is the base for numerous variants, which propose improvement on partial aspects like the estimation of the orientation field. Of course, there are several more hand-crafted approaches to enhance fingerprint samples. For example, there are approaches which aim for a noise reduction or other aspects of preprocessing in more detail [97]. Chapter 4 provides a survey over methods of fingerprint sample enhancement and a benchmark of their effectiveness.

### 3.2.3 Learning-based approaches

Machine learning offers several approaches, which are applicable to fingerprint sample enhancement. There are two sets of approaches, which can be considered state of the art here: dictionary-based approaches and DL approaches.

Cao et al. proposed to use dictionaries for fingerprint sample enhancement [42]. The dictionary consists of small image patches. Each item in the dictionary can be considered a prototypical representative for other similar patches. The dictionary is trained by learning such representative patches. The dictionary shall be able to represent almost all reasonable patches. The workflow for fingerprint sample enhancement is then somehow like solving a jigsaw puzzle: One identifies those candidate items in the dictionary, which are similar to a given patch in a fingerprint sample to be enhanced. In a second step, one combines the candidate puzzle pieces with the neighbouring puzzle pieces. Solving this puzzle for the entire fingerprint sample yields an enhanced sample. This approach was tested on latent fingerprint and achieved quite impressive results.

The approaches in DL can roughly be categorized into two groups. There are direct and indirect enhancement approaches. The former approaches aim directly at an enhancement of the fingerprint sample. The later approaches may generate enhanced fingerprint samples as a by-product.

Direct approaches aim to reconstruct the fingerprint ridge structure. Sahasrabudhe et al. proposed to use Convolutional Deep Belief Networks (CDBN) for this task [251]. Such networks and corresponding trainings are similar to those in Auto-Encoders (see Section 1.1.4). Training can therefore be done in an unsupervised manner. Svoboda et al. proposed to use Generative Convolutional Networks for fingerprint sample enhancement [276]. The idea is to train a network to reconstruct an ideal fingerprint sample from a corrupt sample. For this task, synthetic corrupt and corresponding ideal fingerprints are generated. While the CDBN approach was an unsupervised training approach, this approach is supervised, since it uses ideal fingerprint samples as targets for reconstruction. This approach was also applied to latent fingerprints.

In indirect approaches, by-products may be generated for examples in DL-based approaches to extract minutiae from fingerprint samples. On the one hand side, there are approaches which are inspired by classical minutia extraction. Tang et al. proposed a CNN called *FingerNet* [282]. The architecture of FingerNet imitates the single processing steps from contextual filtering and minutia extraction. For example, there are dedicated layers for estimation and smoothing of the orientation field as well a selection of an appropriate filtering. Even though all neurons are trainable, they are initialized similar to a hand-crafted approach, e.g. edge detection filters for the first layer of orientation field estimation. FingerNet explicitly aims to generate an intermediate representation called *enhanced sample* in its architecture, which is then processed further. However, this enhanced sample could be used as an input to classical feature extraction. A slightly similar approach was proposed by Nguyen et al., which additionally incorporated dictionary-based methods [220]. On the other hand side, there are approaches, which do not incorporate a sophisticated network architecture and initialization like in FingerNet. Those approaches train a CNN from scratch to extract minutiae from fingerprint samples. It is not unlikely that such approaches also generate an intermediate representation of a fingerprint ridge structure, which might also be used for classical minutiae extraction. At the first glance, using such indirectly enhanced samples for classical minutiae extraction might not seem reasonable, since entire CNNs may produce more accurate and more reliable minutiae than the classical approaches would do on such enhanced samples. But due to issues in the interoperability between different minutiae extractors, it may be reasonable after all to use a classical minutia extraction on such enhanced samples.

## 3.3 Contributions

A survey provides an overview over different types of fingerprint sample enhancement (see Chapter 4) [265]. This survey also provides contributions on the assessment of the impact of fingerprint sample enhancement on biometric performance as well as on biometric utility metrics.

There are two contributions on the actual process of fingerprint sample enhancement. One method describes how to use CNNs for reconstruction of fingerprint ridge structures (see Chapter 5) [257]. This method already inspired new approaches [276]. Another method describes how to train CNNs to enhance fingerprint samples, such that minutia extraction actually improves (see Chapter 6) [258]. This approach therefore aims to improve of the biometric performance more directly than earlier approaches.

# Chapter 4

# Survey on the Impact of Fingerprint Image Enhancement

**Summary** Image enhancement is a common pre-processing step to improve the biometric utility of a fingerprint sample. This work inspects several representative approaches of fingerprint sample enhancement. The impact on biometric performance as well as on biometric quality is assessed.
This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was published in the journal *IET Biometrics* in 2018.

[265] P. Schuch, S. Schulz, and C. Busch. Survey on the impact of fingerprint image enhancement. *IET Biometrics*, 7:102–115(13), March 2018. ISSN 2047-4938. URL `http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2016.0088`

**Abstract** The performance of fingerprint comparison algorithms depends on the reliability and accuracy of the features extracted from the fingerprints. The accuracy of the feature extraction algorithms is assumed to depend on the quality of the fingerprint images. Especially low-quality images can be challenging for feature extraction algorithms. Image enhancement may allow to extract features more accurately. There is a lack of extensive and quantitative evaluation of image enhancement methods. This work investigates the impact of seven typical image enhancement methods on biometric sample quality and on biometric performance. The interrelation of image quality and biometric performance is investigated on fourteen datasets. Biometric quality measures are estimated based on image quality metrics NFIQ1 and NFIQ2.0. Biometric performance is tested using MINDTCT and FingerJetFX for feature extraction and BOZORTH3 for biometric comparison. This work shows that the biometric performance can be improved by image enhancement. The significance of improvements depends on both the quality of the datasets and the feature extraction. Thus, there is no single best improvement algorithm. A correlation of changes in scores and image qualities can only be found on the level of entire datasets. No significant correlation can be found for single biometric comparisons.

## 4.1 Introduction

Biometric recognition in general is defined as the "automated recognition of individuals based on their behavioural and biological characteristics" [122]. Fingerprints are one of the most relevant biometric traits used for biometric recognition. Fingerprints have been used for verifying or identifying an individual for decades. Despite the maturity of the fingerprint recognition technology, false positive and false negative errors still occur.

In the process of fingerprint recognition two fingerprint samples are compared to each other. There are different modes to acquire fingerprints samples: capturing with dedicated devices (e.g. live-scanners), ink-based acquisition on paper, and acquisition of latent fingerprints as forensic evidence. Recently it has been proposed to capture fingerprint samples with smartphone cameras or other contact-less photo capture devices. Such modes of acquisition intrinsically influence the properties of the captured fingerprints. Such influences are for example the resolution of the live-scanners or the rolling of the finger over the paper for ink-based acquisition.

Comparison of two fingerprints is usually not performed directly on the fingerprint images. It is done on biometric features extracted from the images. Therefore, fingerprint recognition consists of two separate processes: *biometric feature extraction* and *biometric comparison*. The accuracy of the biometric comparison depends on both processes. The biometric feature extraction process extracts unique features from the biometric samples. Such biometric features are fingerprint minutiae from the fingerprint images for example. The extracted biometric features of two biometric samples are then compared by the biometric comparison process. The biometric performance is assumed to depend on the quality of the biometric samples. A quality measure must express the biometric sample's suitability for biometric recognition. In this understanding, a successful biometric recognition actually depends also on a third component: the quality of the fingerprint image itself.

Methods of image enhancement might increase the biometric sample quality. This in turn might increase biometric performance. Biometric sample quality and performance should be the main aspects when assessing a method for fingerprint image enhancement. But most surveys on fingerprint image enhancement do not perform any assessment at all [74][209][208][146][2][254][238]. Only three publications evaluated biometric performance for a single biometric feature extraction considering at most three datasets [304][8][73]. Only a single survey assesses the influence on biometric performance and biometric quality in terms of NFIQ1 on a single dataset [158]. In contrast, this work provides extensive and quantitative evaluation based on distinct figures of merit. Fourteen datasets with different properties were used for the evaluation. A set of seven representative methods of image enhancement was evaluated. The accuracy achieved with publicly available feature extractors `MINDTCT` [309] and `FingerJetFX` [65] and the comparator `BOZORTH3` [309] was evaluated for the assessment of biometric recognition performance. This publication is the first to adopt NFIQ2.0 along with NFIQ1. These image quality metrics were designed with machine learning from matching scores obtained using BOZORTH3. For the first time, fingerprint image enhancement is evaluated with regard to the correlation between biometric performance and image quality. The figures of merit in this work were designed especially to assess biometric performance and fingerprint quality metrics. Recognition performance is composed of a multitude on single comparison scores. The proposed figure of merit is a single scalar for all comparison scores of those comparisons which are most relevant for biometric performance. These scalars allow to assess the impact of image enhancement of biometric performance. Similar scalars were proposed to focus on the most important samples with respect to biometric quality.

The rest of the publication is organized as follows: The relation to previous work is described in Section 4.2. Section 4.3 categorizes methods of image enhancement and provides a description of typical representatives which are to be assessed. Protocols, datasets, and tools for the evaluation of image enhancements in terms of biometric image quality and biometric comparison performance are described in Section 4.4. Section 4.5 lists the corresponding results. Conclusions are made in Section 4.6, while Section 4.7 adds remarks on the findings of this work.

## 4.2 Related Work

The most related work is a survey by Klir. He evaluated image enhancements on a single dataset for improvement in biometric performance. He correlated the results with NFIQ1 [158].

Furthermore, there are three surveys assessing image enhancement only with regard to biometric performance. Aurora and Garg evaluated seven methods in terms of Equal Error Rate (EER) and timing on a single dataset [8]. Esan et al. evaluated a few methods of image enhancement for their impact on recognition performance [73]. Wang et al. measured the biometric performance in terms of EER for five methods of image enhancement on dataset FVC2000 DB1 [304].

There are more surveys without any assessment at all. Ezhilmaran and Adhiyaman's review study is one of the most recent and broadest surveys which categorize of methods [74]. Misra et al. provide two reports on image enhancement for fingerprints [209][208]. One of them focuses on filtering techniques [209]. Kaur and Kaur, Abbod et al., Sawant and Deore, and Rajin and Ajith only listed published work on fingerprint image enhancement [146][2][254][238].

Baig et al. gave an overview on publications with image enhancements which claimed performance improvement in terms of fingerprint recognition [12]. Yao et al. proposed a quality metric for fingerprints which is alternative to NFIQ2.0 [320].

This work exceeds the related works by putting the image enhancement methods to the test. Seven methods representing different types of theoretical models are tested on their impact on biometric performance and biometric sample quality. Extensive and quantitative evaluations on fourteen datasets were conducted. The impact of each tested method was assessed for two combinations of biometric feature extraction and biometric comparison algorithms. This work also assessed methods of image enhancement by means of NFIQ2.0 and the correlation of biometrics sample quality and biometric performance for the first time.

## 4.3 Methods of Image Enhancements

A lot of ideas have been proposed to enhance fingerprint images. We grouped the different approaches to fingerprint image enhancement according to their underlying motivation in terms of signal and/or noise model. We then chose to evaluate one common representative for each group. All but one of those representative methods are mentioned in the "Handbook of Fingerprint Recognition" [203]. We identified six groups of models:

- *Signal domain models*: These models are based on the statistics of function values, i.e. the grey-values. One aim is e.g. to stretch the contrast of the signal (see Section 4.3.1).
- *Energy models*: These approaches are motivated by physics. The energy of the signal is modelled to normalize the stimulus on the processing system (see Section 4.3.2).
- *Noise statistics models*: Such approaches use models for the signals' statistics to differentiate the relevant signal from noise (see Section 4.3.3).
- *Frequency domain models*: Modelling in frequency domain can be useful, because the ideal fingerprint ridges may be assumed to be similar to a cosine (see Section 4.3.4).
- *Fingerprint models*: Such models include domain-specific knowledge, e.g. orientation and frequency information (see Section 4.3.5).
- *Compositional models*: These approaches try to model fingerprint samples as a composition of many sub parts or of multiple models, e.g. one model for the signal and one model for the disturbance. Image are decomposed into those parts (see Section 4.3.6).

Each representative is described in its own paragraph. Figure 4.1 allows the comparison of the evaluated methods applied to three samples from dataset FVC2000 DB1. The leftmost column show the original samples and each subsequent column shows an enhanced version of the original images.

The assessed methods are only representatives for their group of methods. A lot of approaches were proposed in the past. The following listing provides an overview on approaches published since 2013 which were not covered entirely in the cited surveys (see Section 4.2).

Others signal domain models are used e.g. in approaches of Bouaziz et al. who proposed to use the meta-heuristics *cuckoo search algorithm* and *bat algorithm*

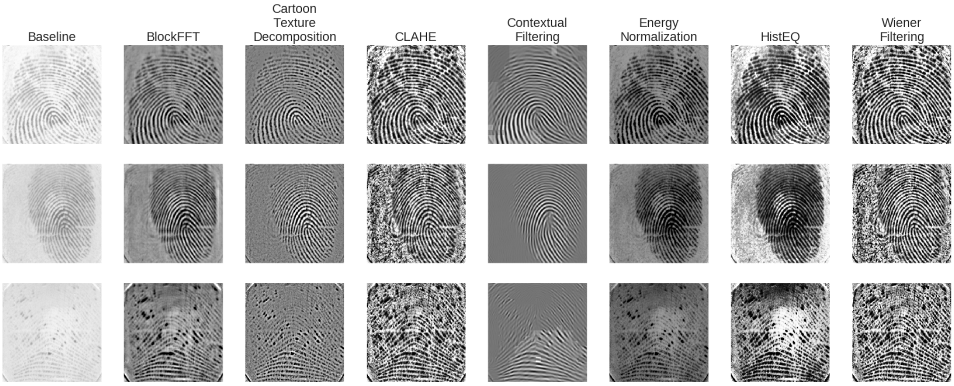


**Figure 4.1:** Three non-enhanced samples are shown for demonstration (leftmost column). Each column is an evaluated enhancement method applied to three samples.

for manipulation of the grey-value histogram [30][31]. Ghosh et al. inspected the application of *cuckoo search algorithm* for fingerprint enhancement, too [89]. Mahashwari et Asthana proposed a fuzzy approach working on the grey-values [198]. Iloanusi proposed an adaptive and data-driven thresholding [115]. Stephen et al. and Stephen proposed two variations of adaptive filtering using particle swarm optimization for parametrization [274][273]. Kabir et al. and Kabir proposed a cascade of three different filter types [144][143].

Noise statistics models are e.g. used by Selvi et George. They proposed a fuzzy approach with adaptive thresholding to deal with noise [266]. Hari et al. proposed to apply quadratic edge filters to deal with noisy background [105].

Several more frequency domain models were proposed, too. Neethu et al. proposed a variation on BlockFFT using 4x4 blocks [219]. Tarar et Kumar proposed to use FFT iteratively for enhancement [283]. Lee et al. proposed an approach for filtering in frequency domain while taking the irregularities in orientation at the locations of singular points and minutiae into account [174]. Deshmukh et al. and Wang et al. proposed a wavelet-based image enhancement. The latter evaluate by improvement in fingerprint classification [64][299]. Bandur et al. proposed to use Log-Gabor filters in frequency domain [16]. Sutthiwichaiporn et al. proposed an iterative approach of filtering in frequency domain and starting iteration in high quality areas [275]. Borra et al. proposed to use *Atom wave transform* with hard tresholding in the transform domain for de-noising [28].

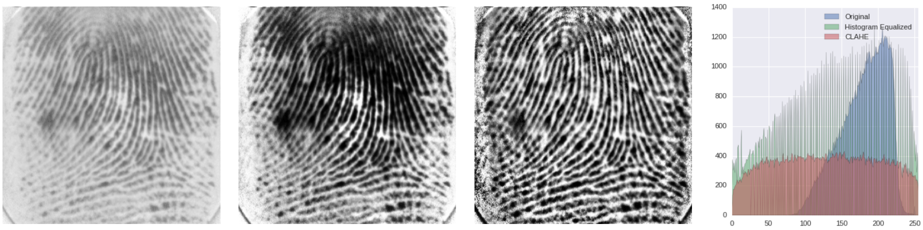
Contextual Filtering is base for numerous variants. Bartåtuněk et al., Rao et al., and Geng et al. proposed a adaptive variation of Contextual Filtering [17][239][83]. Fahmy et Tabet proposed to estimate the local frequency by Radon transformation [75]. Mohammedsayeemuddin et al. proposed a computational less complex variation [210]. Nilam et Joshi propose to performed the Contextual Filtering in frequency domain [221]. Kočevar et al. proposed a block-wise preprocessing before Contextual Filtering [159]. Gahfooor et al. and Zahedi et Ghadi combined Contextual Filtering with filtering in frequency domain [87][325]. Yang et al. proposed a variation of this, which is adaptive to the input [317]. Khan et al. proposed to use multi-scale directional filter bank for the orientation and frequency estimation [152]. Ahmed et al. proposed to use Discrete Curvelet Transform before Contextual Filtering [6]. Divya and Chauhan proposed some tweaks for improving Contextual Filtering [66][55]. Dealing with latent fingerprints is particularly challenging. Baig et al. emphasized the importance of segmentation before application of Contextual Filtering to latent fingerprints [13]. Ahmad et al. proposed to divide latent fingerprints into regions on which morphological operations are to be applied later [5]. There were more approaches proposed which use fingerprint models but which are no variation of Contextual Filtering. Khan et al. and Abdallah et al. proposed to use anisotropic diffusion as fingerprint images are made of flow-like structures and therefore assumed to be anisotropic [153][3]. Gottschlich proposed the usage of curved filters and to make use of knowledge about fingerprints [94]. This idea is picked up again by Khachay et Pasynkov and Mei et al. [149][206]. Sharma et al. proposed to use vortex filters which take into account the characteristic of ridges to be directed edges [267]. Cătălin evaluated various filtering techniques to achieve skeletonized ridge image [53].

Many interesting approaches were proposed in the domain of compositional models. Feng et al. proposed to use local dictionaries of orientation patches for later filtering [77]. Cao et al. and Jain et Cao used dictionaries of local ridge patches [42][133]. Jain et al. used dedicated dictionaries for application to children [136]. Wang et Liu and Liu et al. proposed to use dictionaries of ridge like patches [300][190]. Kumar et Velusamy proposed to learn dictionaries of orientation patches for later use during enhancement [167]. Schuch et al. proposed to train and apply De-Convolutional Auto-Encoders for fingerprint enhancement [257]. Khan et Khan proposed to use a data driven approach to reassemble a fingerprint from its decomposition into directional images [151]. Liu et Zhang used a model of the finger to assemble fingerprint from multiple 3D images [186].

Testing other methods is out of scope of this work. Combinations of methods are possible, but also out of scope.

### 4.3.1 Signal Domain Models: (Localized) Histogram Equalization

The idea of *Histogram Equalization* is to remap the intensities in such a way that the intensity distribution spreads over the resulting histogram as uniform as pos-



**(a)** Input image **(b)** After HistEQ **(c)** After CLAHE **(d)** Grey-values

**Figure 4.2:** Histogram Equalization distributes the intensities as equal as possible over the entire range of intensities.

sible. The order of the intensity values shall be kept the same. This approach uses a model for the signal domain to achieve this. A monotonic function is used to remap the intensity values. This transfer function $T(i)$ for each grey-value intensity $n$ can be derived directly from the empirical cumulative density function CDF$(n)$:

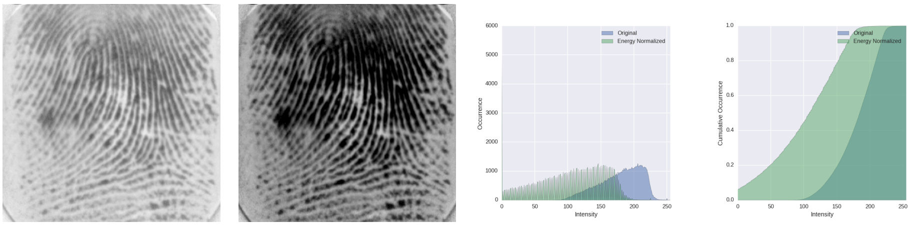
$$T(n) = \text{CDF}(n) * 255 \tag{4.1}$$

Figure 4.2 shows the enhancement of an image. By such a mapping, the intensity values are spread over the entire histogram (see Figure 4.2d) and the contrast gets stretched over the entire image. This algorithm has the intrinsic trait that different grey-values may be mapped to the same grey-value. Thus, after enhancement less grey-value bins might be used than before. This can be observed in Figure 4.2d by inspecting the histograms and its empty bins.

While Histogram Equalization is applied to an entire image, *Contrast Limited Adaptive Histogram Equalization* (CLAHE) stretches the contrast in a more local manner [336]. This overcomes limitations of the simple Histogram Equalization. Figure 4.2c shows the enhancement of the sample image.

Comparing Figures 4.2 b and c reveals the local adaptivity of CLAHE. CLAHE can e.g. stretch the low-contrast area in the upper half of the sample fingerprint while Histogram Equalization cannot. Unfortunately, CLAHE is not sensitive whether or not the relevant signal in present locally. Therefore, every local signal including noise is stretched. This is most obvious in the background of the fingerprint images: CLAHE amplifies the noise in this area, too, while simple Histogram Equalization aplifies only little here.

For both methods implementations from `opencv` were used [32].



**(a)** Input image **(b)** Enhanced image **(c)** Grey-values **(d)** Cumulative

**Figure 4.3:** Energy Normalization shapes the mean and the standard deviation of the grey-value distribution to distinct target values. The image gets thereby specific characteristics. All enhanced images theoretically have the same amount of energy.

### 4.3.2 Energy Models: Energy Normalization

Hong et al. proposed a reshaping of the grey-value distribution of the fingerprint image data. This is part of a larger enhancement workflow [110]. This method was evaluated in two ways: stand-alone and as a part of *Contextual Filtering* approach (see Section 4.3.5). The aim of the distribution shaping of grey-values is to let the mean and the standard deviation of the enhanced image meet distinct target values. The remapping function $m(g)$ operating on a grey-value intensity $g$ is defined according to the following equation:

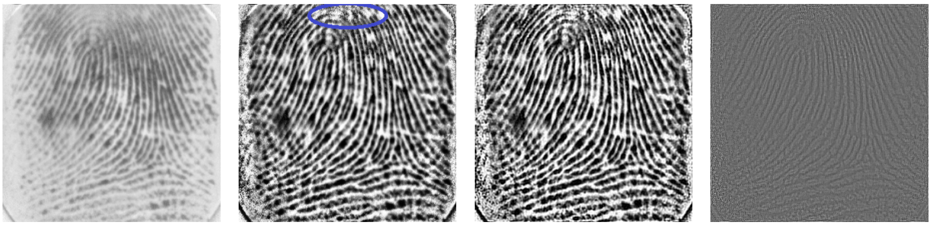
$$m(g) = \begin{cases} \mu_T + \sqrt{\frac{\sigma_T^2}{\sigma_S^2}(g - \mu_S)}\text{ , if } g > \mu_S \\ \mu_T - \sqrt{\frac{\sigma_T^2}{\sigma_S^2}(g - \mu_S)}\text{ , else} \end{cases} \tag{4.2}$$

where $\mu_S$ and $\mu_T$ are the mean of the source and target grey-value distribution while $\sigma_S^2$ and $\sigma_T^2$ are the corresponding variances of both distributions. By doing so, distributions of grey-values in the different enhanced images get more similar to each other. Most important is the aspect that the resulting images should all have similar energy. By doing so, the stimulus on the processing system is normalized.

Hong et al. propose to choose parameters $\mu_T = 100.0$ and $\sigma_T = 100.0$. Figure 4.3 illustrates the enhancement with the proposed parameters applied.

### 4.3.3 Noise Statistics Models: Wiener Filter

Greenberg et al. proposed to use Wiener Filters for fingerprint image enhancement [97]. Like Energy Normalization, this is only a single processing step in a larger workflow. This workflow aims for a binarization of fingerprint images. This binarization could be used more of less directly for a feature extraction. Such an



**(a)** Input image **(b)** CLAHE **(c)** Wiener Filtering **(d)** Estimated noise

**Figure 4.4:** Wiener filtering tries to reconstruct an image containing a noise-free fingerprint sample from a noisy one - the noise estimated is best visible in the center of the top of the fingerprint (blue ellipse in (b)).

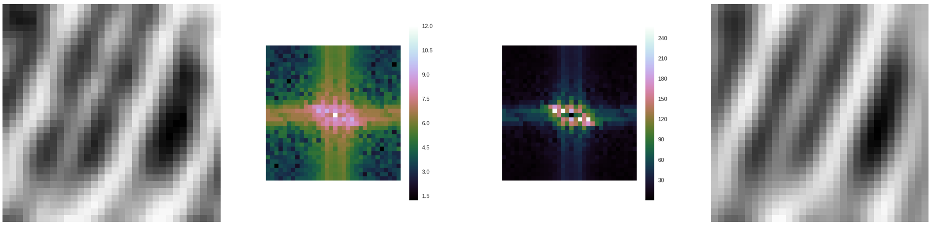
enhancement is out of scope because an assessment with image quality metrics NFIQ1 and NFIQ2.0 would not be reasonable. Thus, it was evaluated stand-alone only.

$$\min_{\mathcal{W}} \mathbb{E}\left\|\underbrace{B_{\text{ideal}}}_{\text{Ideal}} - \mathcal{F}^{-1}(\underbrace{\mathcal{W}}_{\text{Filter}} \cdot \underbrace{\mathcal{F}(\overbrace{B_{\text{ideal}}}^{\text{Ideal}}) \cdot \overbrace{H}^{\text{Degradation}} + \overbrace{\mathcal{N}}^{\text{Noise}})}_{\mathcal{F}(\text{ReceivedImage})}\right\|^2 \tag{4.3}$$

The received image is assumed to be composed of an ideal image $B_{\text{ideal}}$ and additive Noise $\mathcal{N}$. The ideal image is corrupted by a degradation function $H$. The idea is to estimate the statistics of the ideal signal and of the noise. The estimation is based on the received signal. Finally, a filter $W$ is to be designed to recover ideal image $B_{\text{ideal}}$ from the received image. This approach assumes that the noise is additive and that the auto-correlation of the noise is known. Figure 4.4 visualizes the application of an unsupervised variant of the Wiener filtering which is provided by `opencv` [32].

### 4.3.4 Frequency Domain Models: Block-wise Fourier Spectrum Enhancement

Watson et al. proposed an enhancement in the frequency domain using a *Block-wise Fourier Spectrum Enhancement* (BlockFFT) [305]. The idea is that the signal of a fingerprint is locally similar to an oriented 2D-cosine. If so, the signal should be observable in the frequency domain. That specific frequencies can then be enhanced in the frequency domain more easily than in the spatial domain. Watson et al. proposed the *Fast Fourier Transformation* in order to transfer a small block $B$ (for locality) from time domain to frequency domain. The corresponding Fourier



**(a)** Input block **(b)** Log. spectrum **(c)** Scaling **(d)** Enhanced block

**Figure 4.5:** Fingerprint images can be enhanced in the Fourier Spectrum.

Spectrum $\mathcal{F}$ is amplified relative to its Power Spectrum $|\mathcal{F}|$:

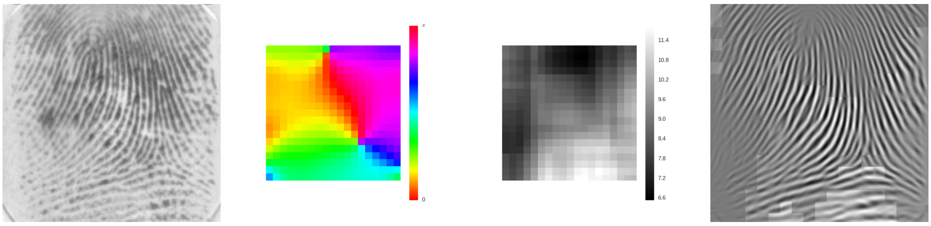
$$B_{\text{Enh}} = \mathcal{F}^{-1}(\underbrace{|\mathcal{F}(B)|^k}_{\text{Scaling}} \mathcal{F}(B)) \tag{4.4}$$

with $k$ proposed to be 0.4. Frequencies with more energy get thereby enhanced more than those with less energy. Assuming that the fingerprint ridge frequency itself is the strongest contributor to the spectrum, this signal will be enhanced the most.

Figure 4.5 visualizes the enhancement applied to a single block with the parameters suggested by Watson et al.. This enhancement may lead to artefacts especially at the borders of each image block. These artefacts are arising mainly for two reasons. First, FFT assumes that the signal periodically repeating. In general, this assumption does not hold. Second, the signal's orientation and frequency vary from one block to its neighbouring blocks. To suppress this effect Watson et al. proposed to apply this enhancement to overlapping blocks. The overlaps are merged later. A block size of $64\,\text{pixels} \times 64\,\text{pixels}$ is proposed for images with a resolution of $500\,\text{dpi}$. Blocks are supposed to have an overlap of $16\,\text{pixels}$.

### 4.3.5 Fingerprint Models: Contextual Filtering

Energy Normalization (see Section 4.3.2) is only a part of an approach proposed by Hong et al. [110]: Maltoni et al. called it *Contextual Filtering* [203]. In contrast to most of the other approaches, contextual filtering is especially tuned for fingerprint image enhancement. It focusses on the fingerprint's characteristics, e.g. ridge orientation and frequencies. Thus, this approach uses specific knowledge about the domain of fingerprints. First, Hong et al. suggested to normalize the energy (see Section 4.3.2). Afterwards, the local orientation of the ridges is estimated (see Figure 4.6b). Based on the local orientation, the local ridge frequency is estimated



**(a)** Input image **(b)** Orientation **(c)** Ridge distance **(d)** Enhanced image

**Figure 4.6:** Contextual filtering extracts an estimation for local orientation ranging from 0 to $\pi$ (b) and local distance between two ridges (c) to apply local Gabor filtering. Errors in each of the estimations degrade the enhanced image significantly.

(see Figure 4.6c). Finally, local orientation and frequency are used to apply locally adapted Gabor filters. The Gabor filters use model driven information from a vicinity to enhance the image. The clarity of the fingerprint ridge structure can therefore be enhanced (see Figure 4.6).

The effect of this enhancement is highly sensitive to the estimations of local orientation and frequency. Both were estimated as described by Hong et al. [110].

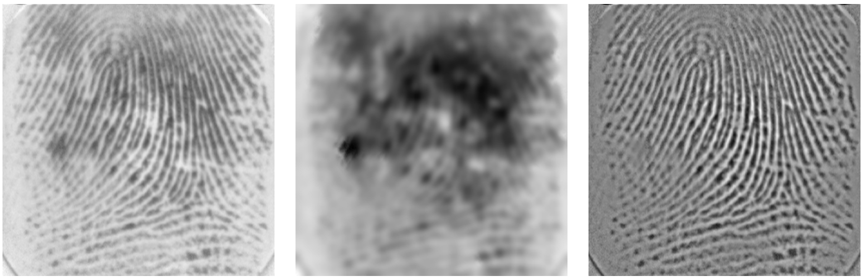
This approach is likely to be *"the most widely used technique for fingerprint image enhancement"* [203]. It is the base for numerous variants with modified working steps, e.g. estimation of local orientation and frequency by using Fourier transformation.

### 4.3.6 Compositional Models: Cartoon-Texture-Decomposition

Buades et al. proposed a decomposition of the fingerprint image into two components [35]. Only the texture component is assumed to contain the relevant information of the fingerprint pattern and will be processed further. The cartoon component is not relevant. This enhancement is a common pre-processing step in the feature extraction workflow for fingerprint recognition - especially when processing latent fingerprints [329][324][42].

The idea is to measure sensibility regarding smoothing operations. The so-called Local Total Variation (LTV) is used as a measure for this. It is defined as follows:

$$\mathrm{LTV}_\sigma(f)(x) = L_\sigma * |\nabla f|(x) \tag{4.5}$$



**(a)** Input image **(b)** Cartoon component **(c)** Texture component

**Figure 4.7:** Local Total Variation can be used to decompose an image (a) into a so-called *cartoon* (b) and a *texture* (c) component.

LTV describes the amount of variation of a function $f$ in a region around a point $x$. A measure $\lambda_\sigma(x)$ can be constructed to describe how much variation remains after a smoothing with some low-pass $L_\sigma$

$$\lambda_\sigma(x) = \frac{\mathrm{LTV}_\sigma(f)(x) - \mathrm{LTV}_\sigma(L_\sigma * f)(x)}{\mathrm{LTV}_\sigma(f)(x)} \tag{4.6}$$

The measurement $\lambda_\sigma(x)$ is in the range from 0 to 1. Finally, $\lambda_\sigma(x)$ can be used to generate a *cartoon* component $u$ and texture component $v$:

$$u(x) = f(x) + w(\lambda_\sigma(x))((L_\sigma * f)(x) - f(x)) \tag{4.7}$$

$$v(x) = f(x) - u(x) \tag{4.8}$$

where $w : [0,1] \mapsto [0,1]$ is an additional transfer function which might be applied. Yoon et al. proposed for example a monotone, piece-wise linear transfer function [324].

Figure 4.7 visualizes the decomposition into both components. While the cartoon components represents the component of low frequencies, the texture part contains a clearly visible fingerprint.

## 4.4 Assessment of the Impact on Biometric Quality and Performance

### 4.4.1 Tools

All tools used for evaluation are publicly available. The tools for biometric feature extraction and biometric comparison will be treated as black boxes in this work.

#### Image Quality Metrics

Biometric image quality shall have predictive power on the biometric utility of a biometric sample. This means that the quality of a sample should express the reliability of decisions based on biometric comparisons involving this sample. Biometric comparison quality can be estimated by quality measures on the fingerprint images themselves. The National Institute of Standards and Technology of USA (NIST) developed such a quality metric. It is called *NIST Fingerprint Image Quality* (NFIQ1)[280]. It was developed for optical live-scanners with a resolution of about 500 dpi. NFIQ1 assigns ordinal scores from 1 to 5 to fingerprint image. A score of 1 indicates highest sample quality. A score of 5 indicates worst sample quality. An artificial neural network has been trained on assign the quality classes to fingerprint images.

NIST presented the successor of NFIQ1 called NFIQ2.0 [222]. An NFIQ2.0 score represents the probability of a sample to belong to a group of reliable high-quality samples. The idea is to calculate the probability based on quality-relevant features, e.g. Frequency Domain Analysis, Local Clarity Score, Orientation Certainty Level, Orientation Flow, and Ridge Valley Uniformity. Quality scores range from 0 to 100. 0 represents lowest biometric utility. 100 represents highest biometric utility. The features are estimated blockwise and are therefore local. Each feature per se seems to be reasonable to describe the biometric comparison quality. The potential of NFIQ2.0 lies in the combination of these complementary features. The combination of features was achieved with the method Random Forest from machine learning [223] [227].

#### Feature Extraction

Minutiae features were extracted with two different algorithms: `MINDTCT` and `FingerJetFX`. The former is provided by NIST, the latter by the company DigitalPersona. Both algorithms extract the locations, directions, and types of the minutiae.

**Table 4.1:** Datasets used for evaluation. Characteristics are low-cost sensor (L), strong rotation (R), displacement (D), moisture (M), low quality (Q), and control on the acquisition (C). Data suited best (bold) to evaluate image enhancement is captured with optical live-scanners with a resolution of about 500 dpi and has no intrinsic characteristics which are challenging to the biometric comparison.

| Dataset | Samples | Dimensions | Resolution | Data Origin | Characteristics |
|---|---|---|---|---|---|
| **FVC2000 DB1** | 800 | 300×300 | 500 dpi | Optical | L |
| FVC2000 DB2 | 800 | 364×254 | 500 dpi | Capacitive | L |
| **FVC2000 DB3** | 800 | 478×448 | 500 dpi | Optical | |
| FVC2002 DB1 | 800 | 388×374 | 500 dpi | Optical | R, D, M |
| FVC2002 DB2 | 800 | 560×296 | 569 dpi | Optical | R, D, M |
| FVC2002 DB3 | 800 | 300×300 | 500 dpi | Capacitive | R, D, M |
| FVC2004 DB1 | 800 | 640×480 | 500 dpi | Optical | R, D, M |
| FVC2004 DB2 | 800 | 364×328 | 500 dpi | Optical | R, D, M |
| FVC2004 DB3 | 800 | 480×300 | 512 dpi | Thermal Sweeping | R, D, M |
| FVC2006 DB2 | 1,680 | 560×400 | 569 dpi | Optical | R, D, M, Q |
| FVC2006 DB3 | 1,680 | 500×400 | 500 dpi | Thermal Sweeping | R, D, M, Q |
| **MCYT DP** | 39,600 | 400×256 | 500 dpi | Optical | C |
| MCYT PB | 39,600 | 300×300 | 500 dpi | Capacitive | C |
| NIST SD14 | 54,000 | 768×832 | 500 dpi | Ink-based | |

### Biometric Comparison

The comparison was performed by algorithm BOZORTH3. It is provided by NIST. BOZORTH3 works on fingerprint minutiae. It is compatible with the features extracted by MINDTCT and FingerJetFX.

## 4.4.2 Datasets

The impact of image enhancement on biometric performance and sample quality needed to be evaluated on representative data. It was performed in a manner that results can be reproduced. We selected the appropriate data in three steps:

1. We selected a set of relevant datasets.
2. We selected only those samples from the datasets which contain sufficient information.
3. We selected only those samples from the datasets which are most relevant for the biometric performance.

A multitude of datasets is publicly available. The first step was to select a set of relevant datasets from those available. The benchmark series *Fingerprint Verification Competition* (FVC) has four editions, which supply the following volumes of datasets: 2000 [200], 2002 [201], 2004 [202], and 2006 [47]. All but five datasets from this series were selected for assessment. One dataset with low resolution and all synthetic datasets were excluded. Moreover the *Ministerio de Ciencia y Tecnología* of Spain (MCYT) provided a multi-modal data-set containing two subsets of fingerprint samples [228]: *MCYT DP* and *MCYT PB*. NIST provides a dataset *NIST SD14* [306]. These fourteen datasets represent the diversity of potential data sources and characteristics which may occur in real world application. Table 4.1 gives an overview of the evaluated datasets and their characteristics. Some of the datasets have special characteristics. These characteristics restrict an evaluation by quality metrics NFIQ1 or NFIQ2.0 (see Section 4.4.1). NFIQ1 was designed for application to fingerprint images with a resolution $500\,\mathrm{dpi}$ captured with optical live-scanners only. Thus, only datasets containing images from optical live-scanner should be considered for both image quality metrics. In addition to this, NFIQ2.0 is designed to process even ink-based fingerprints. In other cases the evaluation of recognition accuracy may be compromised. During capturing of most of the datasets the participants were encouraged to enforce strong displacements and rotations of the fingerprint. Both aspects are obviously challenging for the biometric comparison. But an image enhancement method is not a reasonable counter measure to such challenges. Three datasets comply best with these requirements of the image quality metrics: FVC2000 DB1 and DB3, and MCYT DP. NIST SD14 is appropriate for assessment with NFIQ2.0. MCYT BP, MCYT DP, and NIST SD14 contain numbers of samples which are of orders of magnitude larger than the other datasets. All evaluations were performed on only the first ten percent of these datasets. By doing so, such subsets can still be assumed representative for the evaluation and those subsets are still larger than the other sets.

In a secend step only those images were selected which provide sufficient information for biometric recognition. The larger the area of a fingerprint is, the more information is contained in the fingerprint. Therefore, only those images $i$ will be kept where the area $A(i)$ of the fingerprint is large enough to provide sufficient information for comparison. The area $A$ is estimated by one of the features of NFIQ2.0. Let $S$ an entire dataset. The subset $S_{\text{Large}}$ is defined as follows:

$$S_{\text{Large}} = \{i \in S : A(i) >= \mu_{A(S)} - 2\sigma_{A(S)}\} \tag{4.9}$$

where $\mu_{A(S)}$ is the mean over all fingerprint areas in the dataset $S$ and $\sigma_{A(S)}$ the respective standard variation. The intention of this was to avoid genuine comparisons that are attributed with a low comparison score due to a small overlapping region of the image pair. Small overlaps for two fingerprints result in little information, which both sample have in common. The challenge of small overlap is none an image enhancement can take care of. Therefore those samples were discarded from further evaluation.

Finally, the most relevant comparisons were selected. We therefore identified the samples which were involved in comparisons resulting in unexpected high or low comparison scores. Therefore, biometric features were extracted from the remaining non-enhanced samples with `MINDTCT` and `FingerJetFX`. Comparison was performed on both sets of extracted features. Afterwards the comparison scores were inspected. We identified the 2,000 non-mated comparisons resulting in the highest biometric comparison scores (so-called *Highest Impostors*). Then we identified the 2,000 mated comparisons resulting in the lowest biometric comparison scores (so-called *Lowest Genuines*). Highest Impostors and Lowest Genuines formed the fixed set of *critical comparisons*. All samples from dataset $S$ involved in the critical comparisons were denoted as the *critical images* $C(\text{FE}, S)$. These sets depend only on the feature extractor FE but not on the enhancement method.

By doing so, the selection of data for evaluation was completed. The critical comparisons were repeated on features extracted from the enhanced images. This allowed to inspect the impact of image enhancement.

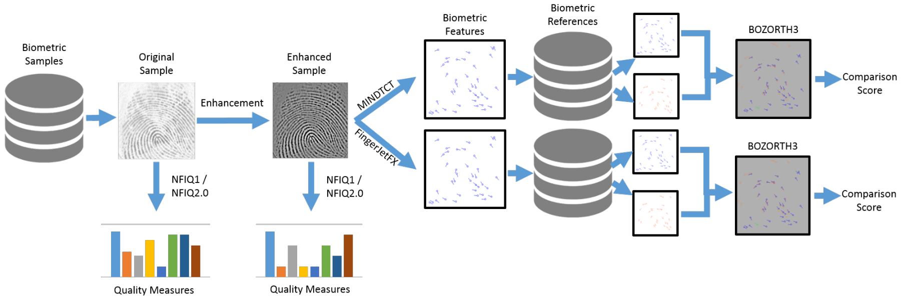


**Figure 4.8:** Workflow for all evaluations

### 4.4.3 Protocols

Figure 4.8 visualizes the workflow which was applied to the samples of the prepared datasets.

#### Protocol 1: Evaluation of NFIQ1

The distribution of NFIQ1 scores is an indicator for the characteristic of the datasets. Scores for NFIQ1 were extracted before and after enhancement on the critical images. This allowed to track the development of the distribution. Samples with NFIQ1 score $\sigma_{\text{NFIQ1}}$ of 3, 4, and 5 are of lowest assumed biometric utility (see Figure 4.9c). The relation of number of samples with NFIQ1 score larger than two after enhancement Enh can therefore be used as an indicator on the quality of a dataset $S$:

$$p_{\text{NFIQ1}}(S, \text{Enh}) = \frac{\#\{i \in S : \sigma_{\text{NFIQ1}}(\text{Enh}(i)) >= 3\}}{\#S} \tag{4.10}$$

Values $p_{\text{NFIQ1}}(S, \text{Enh})$ will be calculated relative to the values before enhancement Enh per dataset $S$ to assess the relative improvements.

### Protocol 2: Evaluation of NFIQ2.0

NFIQ2 scores $\sigma_{\text{NFIQ2.0}}$ were extracted before and after enhancement on the critical images $C(\text{FE}, S)$. This was done for each dataset $S$ and for both feature extractors FE. The distribution of NFIQ2 scores was determined by calculating the empirical cumulative probability density function $\text{CDF}_{S,\text{Enh}}(s_{\text{NFIQ2.0}})$. This curve was characterized by the scalar value of its *Area under the Curve* $\text{AuC}_{\text{NFIQ2}}(S, \text{Enh})$:

$$\text{AuC}_{\text{NFIQ2}}(S, \text{Enh}) = \int_0^{100} \text{CDF}_{S,\text{Enh}}(\sigma_{\text{NFIQ2.0}}) d\sigma_{\text{NFIQ2.0}} \tag{4.11}$$

Values $\text{AuC}_{\text{NFIQ2}}(S, \text{Enh})$ were calculated relative to the values before enhancement per dataset to allow better comparison across datasets.

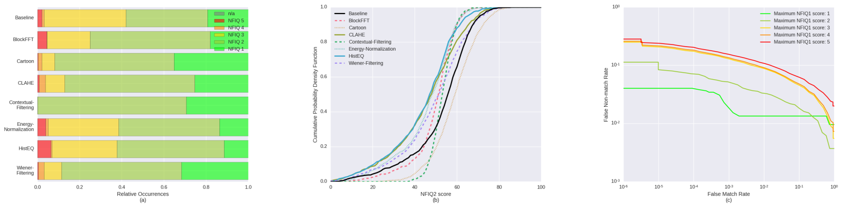
### Protocol 3: Evaluation of biometric performance

All critical comparisons were repeated with features extracted from the enhanced images. *Detection Error Tradeoff* curves (DET) curves were calculated on the resulting comparisons scores. The Area under the Curve $\text{AuC}_{\text{DET}}(S)$ were then calculated for the DET curves. DET curves were in linear scale by intention: as only a small set of mated and non-mated comparisons were inspected, logarithmic scaling would have been misleading. Values were calculated relative to the values achieved before enhancement.

### Protocol 4: Evaluation of predictive power with respect to image enhancements

This evaluation protocol uses results from protocols 1-3. The quality of the biometric comparison is assumed to depend on the qualities of both samples involved. One may assume that the sample with the lower quality has more influence on the comparison than the other sample. We therefore defined the pair-wise quality $\hat{q}_{\text{Enh}}^{\cdot}$ for samples $i_1$ and $i_2$ to be the lower quality of both samples:

$$\begin{aligned} \hat{q}_{\text{Enh}}^{\text{NFIQ1}}(i_1, i_2) &= \max(\sigma_{\text{NFIQ1}}(\text{Enh}(i_1)), \sigma_{\text{NFIQ1}}(\text{Enh}(i_2))) && (4.12) \\ \hat{q}_{\text{Enh}}^{\text{NFIQ2}}(i_1, i_2) &= \min(\sigma_{\text{NFIQ2}}(\text{Enh}(i_1)), \sigma_{\text{NFIQ2}}(\text{Enh}(i_2))) && (4.13) \end{aligned}$$



**(a)** NFIQ1 scores **(b)** NFIQ2.0 scores **(c)** DET curves

**Figure 4.9:** Distribution of NFIQ1 and NFIQ2.0 scores of images from $C(\text{FVC2000 DB1}, \text{MINDTCT})$ and dependence of performance with respect to NFIQ1 scores.

The difference between the pair-wise qualities $\Delta_{\hat{q}^{\text{NFIQ1}}}$ and $\Delta_{\hat{q}^{\text{NFIQ2}}}$ and differences between comparison scores $\Delta_s^{\text{Comp}}$ after and before enhancement were calculated:

$$\Delta_{\hat{q}}^{\text{NFIQ1}}(\text{Enh}, i_1, i_2) \quad = \hat{q}_{\text{Enh}}^{\text{NFIQ1}}(i_1, i_2) - \hat{q}_{id}^{\text{NFIQ1}}(i_1, i_2) \tag{4.14}$$

$$\Delta_{\hat{q}}^{\text{NFIQ2}}(\text{Enh}, i_1, i_2) \quad = \hat{q}_{\text{Enh}}^{\text{NFIQ2}}(i_1, i_2) - \hat{q}_{id}^{\text{NFIQ2}}(i_1, i_2) \tag{4.15}$$

$$\Delta_s^{\text{Comp}}(\text{Enh}, i_1, i_2) \quad = s(\text{Enh}(i_1), \text{Enh}(i_2)) - s(i_1, i_2) \tag{4.16}$$

If the difference indicates that the pair-wise quality increased, one can assume the comparison to be more accurate. In such cases non-mated comparisons should result in lower scores than before enhancement and vice versa for mated comparisons. The correlation of differences of NFIQ scores and differences of comparison scores before and after enhancement is evaluated for the critical comparisons. NFIQ1 and NFIQ2.0 scores are ordinal. Spearman's $\rho(S, \text{Enh}, \text{FE})$ rank-correlation was therefore used to inspect the correlation of $\Delta_s^{\text{Comp}}$ with $\Delta_{\hat{q}}^{\text{NFIQ1}}$ and $\Delta_{\hat{q}}^{\text{NFIQ2}}$. Since changes in scores were expected to be opposing for non-mated and mated comparisons, correlations were inspected separately.

## 4.5 Results

### 4.5.1 NFIQ1

Fingerprint images with an NFIQ1 score of 3, 4 or, 5 assigned to them are assumed to be of low biometric utility. For instance, figure 4.9c shows DET curves for different levels of allowed NFIQ scores for dataset FVC2000 DB1 with features extracted by MINDTCT. Obviously, the performance is significantly lower when samples with NFIQ1 scores of 3 are included in the evaluation than for evaluations on samples with NFIQ1 score of 1 and 2. Adding samples with NFIQ score 4 or 5

**Table 4.2:** $p_{\text{NFIQ1}}(C(\cdot, \text{MINDTCT}), \cdot)$. Lowest values per dataset are marked bold.

| | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT DP | MCYT PB | NIST SD14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (abs) | 0.42 | 0.10 | 0.59 | 0.05 | 0.08 | 0.41 | 0.10 | 0.42 | 0.16 | 0.28 | 0.12 | 0.19 | 0.35 | 0.57 |
| BlockFFT | 0.59 | **0.21** | 0.52 | 0.08 | 0.19 | 0.81 | 0.64 | 0.71 | 0.98 | 0.55 | 0.89 | 0.34 | 0.67 | 0.51 |
| Cartoon | 0.19 | 0.40 | 1.09 | 0.18 | 0.50 | 0.72 | **0.23** | 0.56 | **0.33** | 0.29 | **0.30** | 0.56 | 0.54 | 1.11 |
| CLAHE | 0.31 | 0.60 | 1.22 | 0.26 | 4.05 | 0.97 | 0.79 | 1.36 | 1.82 | 2.62 | 2.50 | 1.41 | 0.91 | 1.02 |
| Contextual-Filtering | **0.00** | 0.70 | **0.05** | **0.00** | **0.00** | **0.11** | 0.35 | **0.03** | 0.62 | **0.04** | 0.39 | **0.00** | **0.24** | **0.30** |
| Energy-Normalization | 0.92 | 0.82 | 1.24 | 0.32 | 1.08 | 1.48 | 1.16 | 1.17 | 1.09 | 1.53 | 1.40 | 1.14 | 1.37 | 1.12 |
| HistEQ | 0.90 | 0.94 | 1.26 | 0.95 | 2.13 | 1.41 | 2.59 | 1.46 | 1.33 | 1.62 | 1.37 | 2.83 | 1.46 | 0.80 |
| Wiener-Filtering | 0.27 | 0.58 | 1.30 | 0.37 | 4.09 | 0.99 | 0.67 | 1.39 | 2.15 | 2.91 | 3.58 | 1.65 | 0.89 | 0.85 |

to the evaluated dataset does not change the performance significantly any more. Samples with NFIQ1 score of 3 or higher can be assumed to be most challenging.

The distribution of NFIQ1 scores varied depending on the image enhancement applied to the fingerprint images. Figure 4.9a shows exemplarily the distributions measured on the fingerprint images from critical comparisons $C(\text{FVC2000 DB1}, \text{MINDTCT})$. Most significant is the change achieved by Contextual Filtering: all samples with NFIQ1 score of at least 3 have a score of at most 2 after enhancement. Table 4.2 gives an overview on the percentages of $C(\cdot, \text{MINDTCT})$ which have an NFIQ score of 3 or higher after the enhancement method applied. Contextual Filtering significantly reduces the percentage of samples which have an assumed low biometric quality. Contrary to this, some methods of image enhancement even increase the percentage of low quality samples. If an image enhancement reduced the percentage of images with bad quality, the biometric performance might probably be increased.

The percentages on NFIQ1 scores of 3 to 5 is lower than ten percent even before image enhancement for datasets FVC2000 DB2 and DB3, and FVC2002 DB2 and DB3. Relative improvements have to be interpreted carefully since since small changes result in large improvements. The lower the percentages $p_{\text{NFIQ1}}(S)$ are, the noisier the measurements may be.

### 4.5.2 NFIQ2.0

Figure 4.9b shows the cumulative occurrence probabilities of NFIQ2.0 score before and after enhancement exemplary for critical images $C(FVC2000DB1, \text{MINDTCT})$. Only the image enhancement method Cartoon-Texture-Decomposition improved the NFIQ2.0 scores of this set compared to the baseline.

**Table 4.3:** $\mathrm{AuC}_{\mathrm{NFIQ2}}(\cdot,\cdot)$. Best values per dataset are marked bold.

| | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT DP | MCYT PB | NIST SD14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (abs) | 47.9 | 42.8 | 62.5 | 36.4 | 45.4 | 74.4 | 41.5 | 62.1 | 53.6 | 61.7 | 52.5 | 54.7 | 68.6 | 61.8 |
| BlockFFT | 1.07 | **0.91** | 0.91 | 1.20 | 0.97 | 0.73 | 1.13 | 0.84 | 1.12 | 0.86 | 1.14 | 1.01 | 0.85 | 0.94 |
| Cartoon | **0.87** | 1.21 | 0.81 | 1.20 | **0.79** | 0.79 | 0.99 | 0.87 | 0.84 | **0.64** | 0.89 | 1.07 | 0.91 | **0.88** |
| CLAHE | 1.15 | 1.34 | 1.00 | **0.96** | 1.17 | 1.02 | **0.94** | 1.21 | 1.11 | 1.09 | 1.21 | 1.12 | 1.09 | 0.93 |
| Contextual-Filtering | 0.99 | 1.06 | **0.73** | 1.30 | 0.94 | **0.63** | 1.12 | **0.75** | **0.82** | 0.68 | **0.82** | **0.89** | **0.71** | 0.93 |
| Energy-Normalization | 1.15 | 1.33 | 1.00 | 1.37 | 1.02 | 1.01 | 1.40 | 1.05 | 1.01 | 1.00 | 1.03 | 1.19 | 1.11 | 1.02 |
| HistEQ | 1.18 | 1.43 | 1.04 | 1.49 | 1.06 | 1.06 | 1.64 | 1.11 | 1.04 | 1.02 | 1.09 | 1.45 | 1.18 | 0.95 |
| Wiener-Filtering | 1.08 | 1.00 | 1.04 | 1.37 | 1.25 | 0.89 | 1.06 | 1.19 | 1.10 | 1.16 | 1.29 | 1.07 | 0.99 | 0.97 |

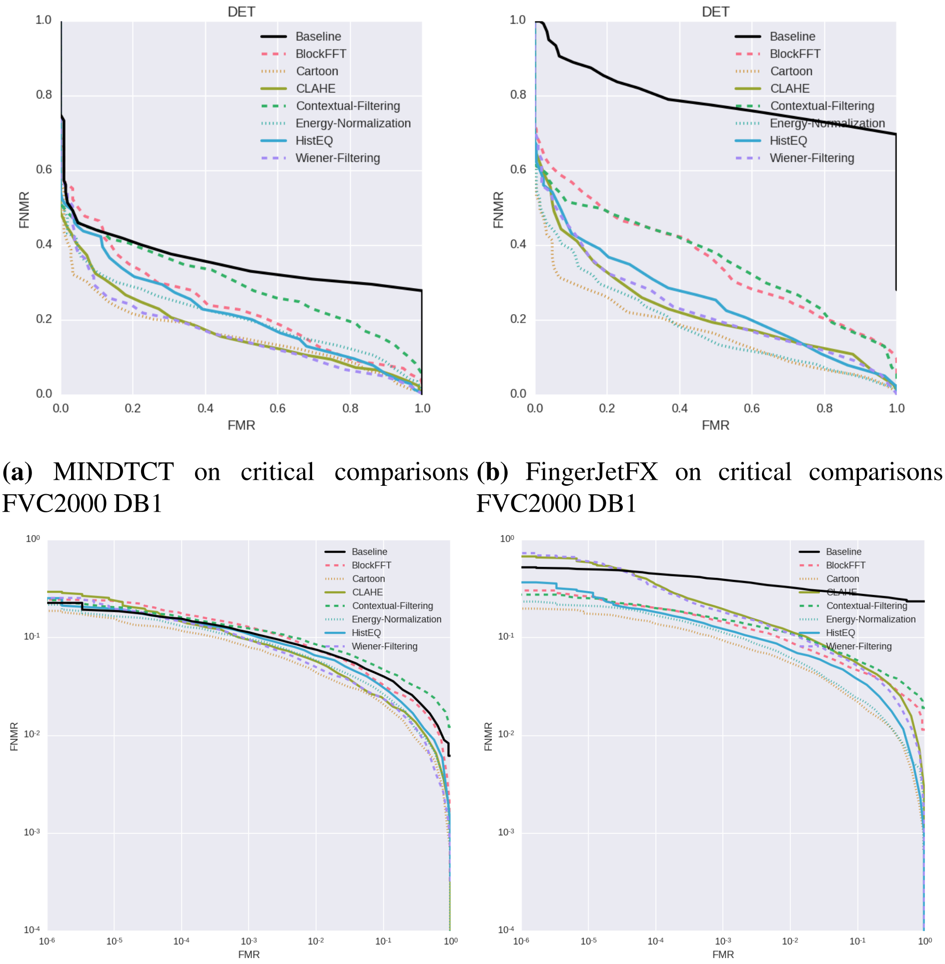
To allow comparison over all datasets and image enhancements $\mathrm{AuC}_{\mathrm{NFIQ2}}(S, \mathrm{Enh})$ was calculated. Table 4.3 summarizes the results. The best improvements in terms of this metric were achieved by the image enhancement methods Contextual-Filtering, Cartoon-Texture-Decomposition, and CLAHE. The methods Cartoon-Texture-Decomposition and Contextual-Filtering resulted in an AuC values which are smaller than values for the baseline on almost all datasets. However, methods Energy Normalization, HistEQ, and Wiener Filtering achieved an AuC larger than the baseline for all datasets. No obvious trend can be found for methods CLAHE and BlockFFT.

### 4.5.3 Biometric Performance

Figure 4.10 visualizes the DET curves before and after image enhancements on FVC2000 DB1 for the critical comparisons. To allow comparability between different image enhancement methods on different datasets, the calculated DET curves are projected into a scalar value: the AuC. Tables 4.4 and 4.5 contain the AuC values for all combinations of datasets, image enhancements, and feature extractors.

According to Table 4.4, there is no single best image enhancement. The method to achieve the best improvement in terms of this metric therefore depends on the dataset. The highest count of best results per dataset was found for methods Cartoon-Texture-Decomposition and Contextual-Filtering. However, the method Contextual Filtering even decreased biometric performance in this metric on some datasets. Thus, not all methods worked out of the box for all datasets.

The biometric performance was remarkably lower for features extracted with FingerJetFX than for those extracted with MINDTCT. This does not yield any advantage of MINDTCT over FingerJetFX. The biometric performance does not depend



**(a)** MINDTCT on critical comparisons FVC2000 DB1

**(b)** FingerJetFX on critical comparisons FVC2000 DB1

**(c)** MINDTCT on entire FVC2000 DB1

**(d)** FingerJetFX on entire FVC2000 DB1

**Figure 4.10:** DET curves before and after image enhancement. The performance can be increased by image enhancements. The impact differs between the entire dataset and the critical comparisons only.

**Table 4.4:** $\text{AuC}_{\text{DET}}(\cdot,\cdot,\text{MINDTCT})$. Rightmost column represents evaluation on the entire dataset.

| | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT BP | MCYT DP | NIST SD14 | FVC2000 DB1* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (abs) | 0.33 | 0.26 | 0.67 | 0.16 | 0.12 | 0.67 | 0.76 | 0.69 | 0.68 | 0.40 | 1.00 | 0.87 | 0.99 | 0.52 | 0.02 |
| BlockFFT | 0.53 | 0.36 | 0.46 | 0.59 | 1.40 | 0.40 | 0.64 | 0.56 | 0.59 | 0.43 | 0.72 | 0.16 | 0.70 | **0.40** | 0.86 |
| Cartoon | **0.32** | 0.54 | 0.54 | **0.55** | 0.67 | 0.41 | **0.56** | 0.38 | **0.48** | 0.38 | 0.59 | 0.37 | 0.79 | 0.52 | **0.44** |
| CLAHE | 0.36 | 0.37 | 0.55 | 0.62 | 0.70 | 0.46 | 0.63 | 0.42 | 0.63 | **0.15** | 0.81 | 0.38 | 0.75 | 0.57 | 0.55 |
| Contextual-Filtering | 0.54 | 0.77 | **0.45** | 1.31 | 1.16 | 0.41 | 0.74 | 0.39 | 0.61 | 0.40 | **0.41** | **0.13** | **0.41** | 0.59 | 1.40 |
| Energy-Normalization | 0.49 | 0.74 | 0.71 | 1.03 | 0.72 | 0.49 | 0.80 | 0.59 | 0.73 | 0.31 | 0.94 | 0.45 | 0.90 | 0.67 | 0.56 |
| HistEQ | 0.46 | 0.41 | 0.58 | 0.68 | **0.66** | **0.40** | 0.67 | 0.45 | 0.64 | 0.22 | 0.83 | 0.39 | 0.76 | 0.60 | 0.68 |
| Wiener-Filtering | 0.33 | **0.31** | 0.55 | 0.63 | 0.70 | 0.45 | 0.61 | **0.37** | 0.56 | 0.19 | 0.69 | 0.22 | 0.59 | 0.61 | 0.49 |

**Table 4.5:** $\text{AuC}_{\text{DET}}(\cdot,\cdot,\text{FingerJetFX})$. Rightmost column represents evaluation on the entire dataset.

| | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT DP | MCYT PB | NIST SD14 | FVC2000 DB1* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (abs) | 0.71 | 0.21 | 0.96 | 0.54 | 0.43 | 0.73 | 1.00 | 0.92 | 0.67 | 0.98 | 1.00 | 0.70 | 1.00 | 1.00 | 0.25 |
| BlockFFT | 0.21 | 0.61 | 0.57 | 0.89 | 0.63 | 0.57 | 0.67 | 0.51 | 0.63 | 0.51 | 0.86 | 0.37 | 0.69 | 0.89 | 0.12 |
| Cartoon | **0.11** | **0.55** | 0.61 | 0.78 | 0.58 | 0.47 | 0.66 | 0.46 | 0.68 | 0.47 | 0.97 | 0.45 | 0.77 | 0.87 | **0.04** |
| CLAHE | 0.14 | 0.65 | 0.54 | 0.70 | 0.64 | 0.40 | 0.61 | **0.34** | **0.56** | **0.29** | 0.80 | 0.43 | 0.65 | 0.79 | 0.11 |
| Contextual-Filtering | 0.22 | 1.61 | 0.59 | 1.02 | 0.78 | 0.81 | 0.70 | 0.50 | 0.80 | 0.56 | 0.76 | **0.25** | 0.76 | 0.89 | 0.15 |
| Energy-Normalization | 0.13 | 0.66 | 0.62 | 0.76 | 0.60 | 0.49 | 0.67 | 0.53 | 0.74 | 0.52 | 0.96 | 0.42 | 0.83 | 0.89 | 0.05 |
| HistEQ | 0.17 | 0.63 | 0.57 | 0.66 | **0.58** | 0.45 | **0.53** | 0.43 | 0.66 | 0.48 | 0.83 | 0.28 | 0.71 | 0.80 | 0.06 |
| Wiener-Filtering | 0.14 | 0.57 | **0.52** | **0.64** | 0.63 | **0.39** | 0.57 | 0.34 | 0.56 | 0.31 | **0.74** | 0.39 | **0.61** | **0.79** | 0.10 |

only on the feature extractor. It depends on their combination with a comparison algorithm. However, the relative improvements are therefore larger for FingerJetFX compared to those for MINDTCT.

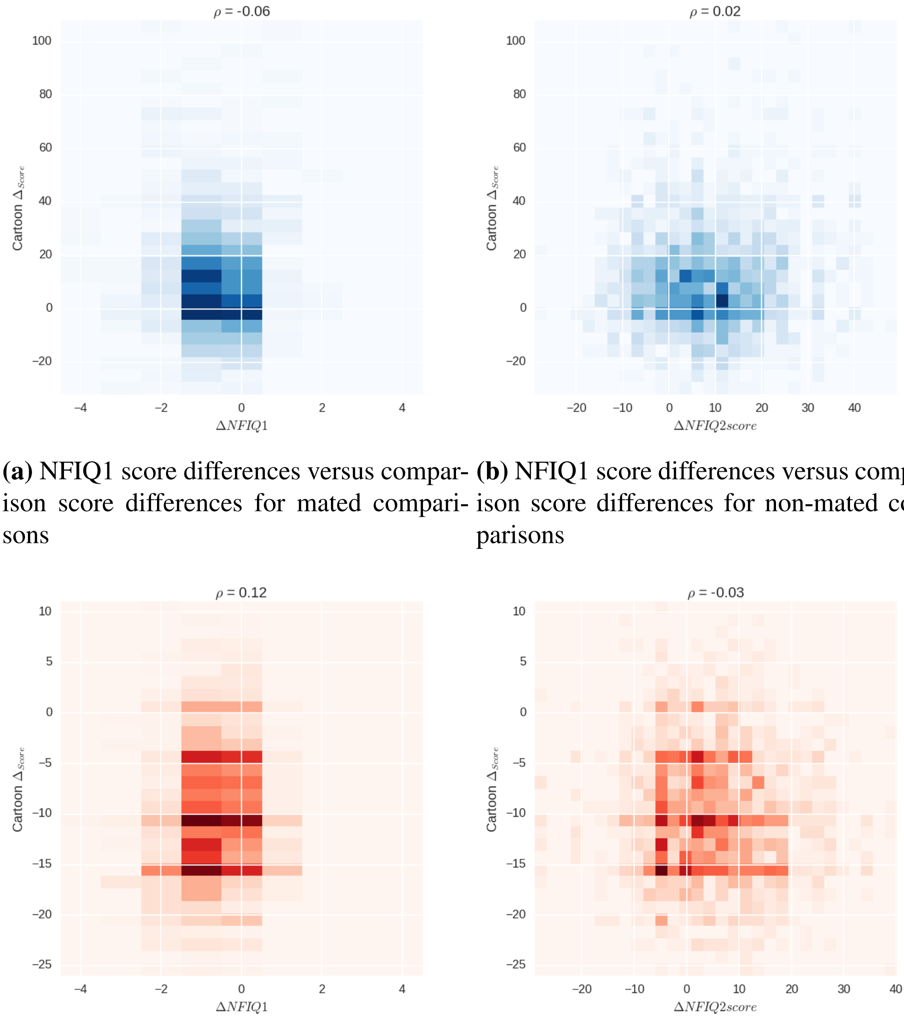
Table 4.5 reveals that no clear trend can be found for the biometric performance based on features extracted with FingerJetFX as well. The highest count of best results was found here for the methods Wiener-Filtering and CLAHE. Again, in few cases the biometric performance was decreased when an image enhancement had been applied to the fingerprint samples. The image enhancement method which achieved best biometric performance therefore does not only depend on the data. It also depends on the feature extractor used.

Both tables include as their rightmost column the result for evaluation on the entire dataset FVC2000 DB1. Values for measures on the entire dataset and on critical comparisons were biased, e.g. by a-priori probabilities of mated and non-mated comparisons. Thus, they were not comparable directly. However, rank-correlation was applicable to evaluate correlation of both measurements. Spearman's rank correlation coefficient is 0.96 when MINDTCT is used and 0.88 when FingerJetFX is used. This indicates a significant correlation. Reduction to critical comparisons for evaluation purposes seems therefore reasonable.

### 4.5.4 Correlation of Image Quality and Biometric Performance

Figure 4.11 shows exemplarily 2D-histograms of the changes in biometric scores and changes in image quality scores when Cartoon-Texture-Decomposition was applied to the critical images $C(\text{FVC2000 DB1}, \text{MINDTCT})$. The visualization is split into mated and non-mated comparisons. Pair-wise qualities were improved by image enhancement. There was a trend for mated comparison to result in higher comparison scores after image enhancement. The trend for comparison scores for non-mated comparison to be lower was even more present. When comparing Figures 4.11b and d, the changes in NFIQ2.0 scores and changes in comparison scores are more like expected for the mated comparisons than for the non-mated ones. A cause for this behaviour may be that NFIQ2.0 has been trained on mated-comparisons. However, no significant correlation between the differences in quality metrics and comparison scores was found at the level of single comparisons.

Nonetheless, the figures of merit for biometric performance and quality are of a certain value. Those figure of merit have units which are hardly comparable. But they provide an order. Thus, rank-correlation can be used to assess the relation of both values. We inspected only those three datasets which are suited best for assessment by NFIQ1. We inspected absolute values for the figures of merit as these were not biased by a baseline measurement. Figure 4.12 visualizes the cor-



**(a)** NFIQ1 score differences versus comparison score differences for mated comparisons

**(b)** NFIQ1 score differences versus comparison score differences for non-mated comparisons

**(c)** NFIQ2 score differences versus comparison score differences for mated comparisons

**(d)** NFIQ2 score differences versus comparison score differences for non-mated comparisons

**Figure 4.11:** Correlation of differences of comparison scores and image quality scores before and after enhancement Cartoon-Texture-Decomposition on critical comparisons (MINDTCT) on FVC2000DB1.

relation for $\mathrm{AuC}_{\mathrm{DET}}$ and $p_{\mathrm{NFIQ1}}$ and $\mathrm{AUC}_{\mathrm{NFIQ2}}$ respectively for features extracted with MINDTCT and FingerJetFX. For MINDTCT we found correlation coefficients of 0.66 for $\mathrm{AuC}_{\mathrm{DET}}$ with $p_{\mathrm{NFIQ1}}$ and 0.72 with $\mathrm{AUC}_{\mathrm{NFIQ2}}$ respectively. This is a fair correlation. For FingerJetFX the correlation coefficients were lower: 0.41 and 0.38. When comparing the subfigures in Figure 4.12, the changes in $\mathrm{AuC}_{\mathrm{DET}}$ for FingerJetFX were smaller than for MINDTCT. This indicated that the image enhancement changed the biometric performance less significant for FingerJetFX. Therefore the correlation over more than one dataset is lower for FingerJetFX than it is for MINDTCT.

When inspecting the figures of merit in detail for Contextual-Filtering, the reduction of $p_{NFIQ1}$ is remarkably. The change is more slightly in $\mathrm{AUC}_{\mathrm{NFIQ2}}$. This reduction is not corresponding to the chance in $\mathrm{AuC}_{\mathrm{DET}}$. Therefore, NFIQ1 seems to be more prone to the over-emphasized ridge structure produced by Contextual-Filtering than NFIQ2.0 seems to be.

Table 4.6 summarizes the evaluated rank-correlation coefficients for all combinations of datasets and image enhancement methods on features with MINDTCT. At the level of single comparisons, both image quality metrics therefore seem to be of limited use for estimation of the impact of an image enhancement on biometric comparison performance. This work treats quality assessment tools, biometric feature extraction, and comparison algorithms as black boxes. They are quite complex. Therefore, reasoning on causes for the behaviour we have found would just be speculation.

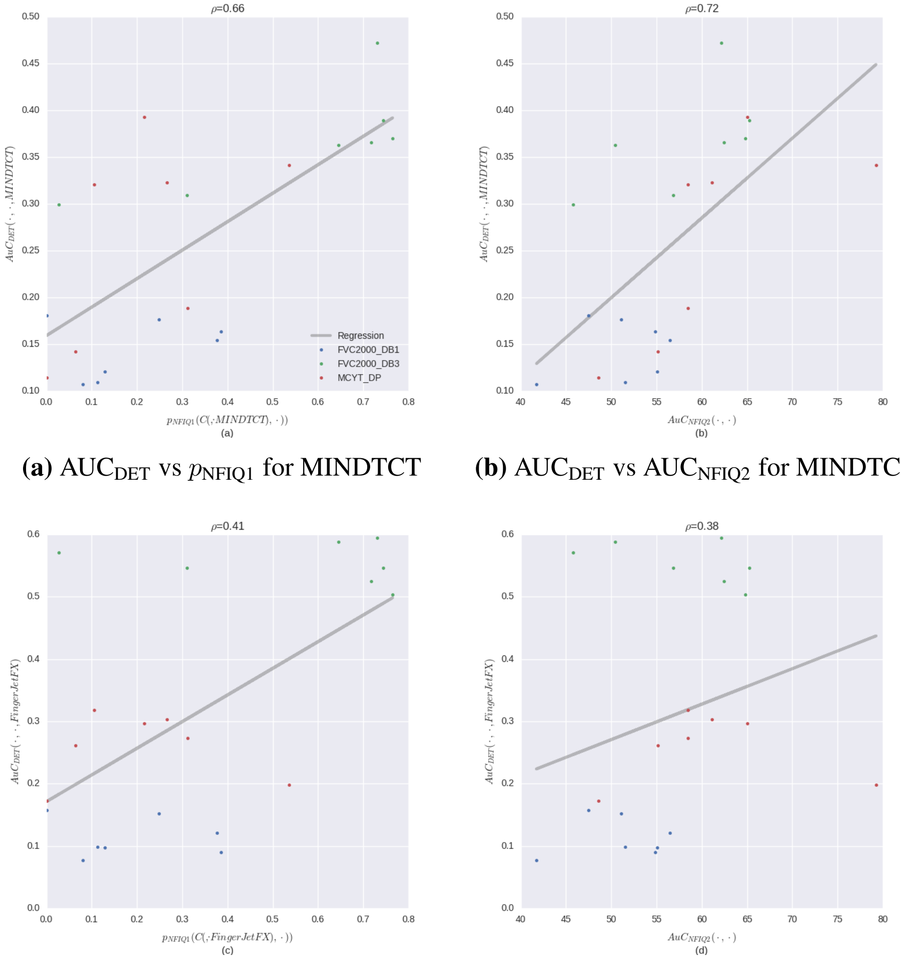
## 4.6 Conclusion

Image enhancement was found to be able to improve the biometric performance in a lot of cases. But the improvement to be achieved varied strongly between datasets. The performance also varies depending on the feature extraction algorithms. Contextual Filtering and Cartoon-Texture Decomposition performed best among the evaluated methods for MINDTCT. Wiener Filtering performed best for FingerJetFX. We found that assessing the impact of image enhancement on the subset of critical comparisons can be an indicator for the performance on an entire dataset. In some constellations of feature extractors and datasets simple image enhancement methods outperformed the more sophisticated approaches. However, no enhancement was able to improve performance for every dataset and for every feature extractor. Thus, no single solution was found to work best out of the box.

In some cases image enhancements improved the biometric quality metrics NFIQ1 or NFIQ2.0 of fingerprint samples. When inspecting sets of images, improvements in terms of quality metrics coincided with improvements in biometric per-

**Table 4.6:** Spearman's $\rho(\cdot,\cdot,M)$ (M = MINDTCT) for differences in pair-wise qualities for NFIQ1/NFIQ2.0 and comparison scores.

| | **FVC2000 DB1** | FVC2000 DB2 | **FVC2000 DB3** | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | **MCYT DP** | MCYT PB | NIST SD14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{BlockFFT}, M)$ | -0.18 | 0.04 | -0.03 | -0.11 | 0.06 | -0.06 | 0.01 | 0.06 | 0.03 | -0.05 | -0.05 | -0.17 | -0.14 | -0.04 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{Cartoon}, M)$ | -0.06 | 0.01 | -0.04 | 0.02 | -0.09 | -0.20 | -0.05 | -0.13 | 0.07 | -0.02 | 0.03 | 0.11 | -0.15 | 0.02 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{CLAHE}, M)$ | -0.05 | -0.08 | 0.03 | -0.06 | 0.02 | -0.11 | -0.05 | -0.08 | 0.15 | -0.08 | 0.02 | 0.03 | -0.12 | 0.04 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{Context.-Filt.}, M)$ | -0.09 | -0.17 | -0.08 | -0.04 | -0.01 | 0.01 | -0.07 | 0.13 | -0.14 | -0.19 | -0.10 | -0.13 | 0.17 | -0.14 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{Ener.-Norm.}, M)$ | -0.04 | -0.02 | -0.14 | -0.00 | -0.10 | -0.02 | -0.03 | -0.07 | -0.01 | -0.02 | 0.02 | -0.02 | -0.02 | -0.01 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{HistEQ}, M)$ | -0.20 | -0.05 | 0.00 | -0.24 | -0.31 | -0.17 | -0.20 | -0.16 | 0.09 | -0.11 | -0.02 | -0.10 | -0.10 | -0.07 |
| $\rho^{\text{Mated}}_{\text{NFIQ1}}(\cdot, \text{Wiener-Filt.}, M)$ | -0.06 | 0.00 | -0.06 | -0.06 | 0.07 | -0.04 | -0.01 | 0.02 | 0.10 | 0.10 | 0.09 | 0.01 | -0.10 | 0.03 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{BlockFFT}, M)$ | -0.03 | 0.04 | 0.27 | -0.09 | 0.01 | -0.11 | 0.03 | -0.04 | 0.10 | -0.05 | 0.02 | -0.10 | 0.05 | 0.05 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{Cartoon}, M)$ | 0.12 | 0.00 | 0.02 | 0.04 | -0.06 | -0.08 | 0.20 | -0.04 | -0.15 | 0.04 | -0.15 | -0.02 | -0.03 | 0.15 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{CLAHE}, M)$ | 0.06 | -0.02 | -0.00 | -0.05 | -0.03 | 0.11 | 0.15 | -0.04 | 0.13 | 0.00 | 0.20 | -0.10 | -0.04 | 0.13 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{Context.-Filt.}, M)$ | 0.06 | -0.09 | -0.10 | -0.02 | 0.14 | -0.17 | 0.12 | 0.15 | -0.00 | -0.35 | -0.26 | -0.11 | -0.05 | -0.02 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{Ener.-Norm.}, M)$ | -0.06 | 0.03 | 0.03 | -0.02 | -0.09 | -0.27 | -0.19 | -0.33 | 0.02 | -0.07 | -0.00 | 0.02 | -0.02 | -0.11 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{HistEQ}, M)$ | -0.12 | 0.01 | -0.06 | -0.12 | -0.29 | -0.23 | -0.07 | -0.13 | 0.08 | -0.14 | 0.09 | -0.19 | -0.07 | 0.12 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ1}}(\cdot, \text{Wiener-Filt.}, M)$ | 0.14 | -0.03 | 0.00 | -0.03 | 0.07 | 0.07 | 0.15 | -0.04 | 0.18 | -0.00 | 0.22 | -0.08 | 0.03 | 0.16 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{BlockFFT}, M)$ | -0.03 | 0.03 | 0.07 | 0.03 | 0.12 | 0.03 | 0.05 | 0.06 | 0.03 | 0.12 | -0.03 | 0.12 | 0.06 | -0.02 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Cartoon}, M)$ | 0.02 | -0.03 | 0.02 | -0.09 | 0.01 | -0.14 | 0.04 | -0.01 | 0.00 | 0.06 | -0.01 | 0.24 | 0.08 | -0.01 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{CLAHE}, M)$ | 0.07 | 0.03 | 0.09 | 0.04 | 0.10 | 0.03 | 0.05 | 0.10 | -0.08 | 0.04 | -0.11 | 0.10 | 0.04 | -0.12 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Context.-Filt.}, M)$ | 0.12 | 0.21 | -0.00 | -0.01 | 0.04 | -0.01 | 0.08 | -0.02 | 0.19 | -0.01 | 0.21 | 0.01 | -0.02 | 0.08 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Ener.-Norm.}, M)$ | -0.06 | 0.07 | 0.08 | -0.01 | -0.05 | 0.08 | 0.07 | 0.01 | -0.02 | 0.02 | -0.02 | -0.03 | -0.02 | 0.01 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{HistEQ}, M)$ | 0.04 | 0.09 | 0.16 | 0.12 | 0.19 | 0.11 | 0.11 | 0.12 | 0.01 | -0.06 | 0.02 | -0.00 | 0.17 | -0.12 |
| $\rho^{\text{Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Wiener-Filt.}, M)$ | 0.04 | 0.04 | 0.02 | 0.06 | 0.06 | -0.00 | 0.05 | 0.07 | -0.04 | -0.08 | -0.01 | -0.07 | 0.03 | -0.05 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{BlockFFT}, M)$ | -0.09 | -0.28 | -0.08 | -0.04 | -0.02 | 0.00 | -0.06 | -0.19 | -0.01 | -0.12 | -0.01 | 0.01 | -0.04 | -0.13 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Cartoon}, M)$ | -0.03 | -0.06 | 0.11 | -0.02 | 0.01 | -0.17 | -0.16 | -0.12 | 0.05 | -0.04 | 0.08 | 0.06 | 0.00 | -0.23 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{CLAHE}, M)$ | 0.12 | 0.16 | 0.08 | -0.04 | 0.04 | 0.02 | -0.09 | -0.00 | -0.15 | 0.01 | -0.13 | -0.10 | 0.12 | -0.18 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Context.-Filt.}, M)$ | -0.05 | -0.06 | 0.27 | -0.07 | -0.06 | -0.02 | -0.15 | -0.15 | 0.37 | 0.08 | 0.36 | -0.02 | -0.03 | -0.15 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Ener.-Norm.}, M)$ | 0.04 | 0.12 | 0.07 | 0.04 | -0.05 | 0.03 | -0.06 | -0.08 | -0.03 | 0.05 | -0.05 | 0.03 | -0.00 | -0.07 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{HistEQ}, M)$ | 0.13 | 0.09 | 0.03 | 0.08 | 0.07 | -0.01 | 0.07 | 0.04 | -0.06 | -0.06 | -0.15 | 0.08 | 0.21 | -0.25 |
| $\rho^{\text{Non-Mated}}_{\text{NFIQ2.0}}(\cdot, \text{Wiener-Filt.}, M)$ | 0.07 | -0.01 | 0.17 | 0.07 | 0.01 | 0.03 | -0.08 | -0.07 | -0.07 | -0.06 | 0.01 | -0.20 | -0.00 | -0.20 |



**(a)** AUC$_{\text{DET}}$ vs $p_{\text{NFIQ1}}$ for MINDTCT **(b)** AUC$_{\text{DET}}$ vs AUC$_{\text{NFIQ2}}$ for MINDTCT

**(c)** AUC$_{\text{DET}}$ vs $p_{\text{NFIQ1}}$ for FingerJetFX **(d)** AUC$_{\text{DET}}$ vs AUC$_{\text{NFIQ2}}$ for FingerJetFX

**Figure 4.12:** The absolute values of the figures of merit and their degree of rank-correlation $\rho$. Each point in the figures is an result from a single method of image enhancement.

formance. This trend was found especially for the non-mated comparisons. The figures of merit introduced in this work were used to estimate the impact of different methods of image enhancement on biometric comparison. But, at the level of single comparison the changes in image quality did not correlate significantly with changes in comparison scores. Neither NFIQ1 nor NFIQ2.0 are therefore suited to assess the impact of image enhancement methods on biometric performance at this level. Actually, none of both was intended to do such an assessment.

All tools used for the assessment were treated as black boxes. Detailed reasoning on the empirically experienced behaviour and correlations would have been nothing more than speculation.

## 4.7 Discussion

Almost all related approaches that were evaluated do have parameters which might be adjusted to the input data and therefore tweaked for a higher performance. Parameters were set as suggested by the references. Arbitrary combinations of image enhancement methods are possible and may also lead to higher performance, too.

Evaluations were performed only on the set of critical comparisons. These are the most relevant cases in terms of biometric performance. We found the reduction to the subset of critical comparisons to be reasonable. But the critical comparisons were not representative by definition. Therefore, performances shall always be cross-checked on the entire dataset.

Two feature extractors were evaluated, but all comparisons were performed with only a single algorithm. Another comparison algorithm might allow inspection on this degree of freedom.

# Chapter 5

# De-Convolutional Auto-Encoder for Enhancement of Fingerprint Samples

**Summary** Auto-Encoders can be trained to reconstruct the relevant ridge structure from fingerprint samples. Synthetic fingerprints and corresponding ideal images are used for training such a model. The trained model enhances fingerprint samples. The impact of the enhancement on biometric performance is assessed on several datasets. This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was presented at the *International Conference on Image Processing Theory, Tools and Applications* in Oulu (Finnland) in 2016.

[257] P. Schuch, S. Schulz, and C. Busch. De-convolutional auto-encoder for enhancement of fingerprint samples. In *6th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–7. IEEE, 2016

**Abstract** Reliability and accuracy of the features extracted from fingerprints are essential for the performance of any fingerprint comparison algorithm. Image Enhancement as a pre-processing step allows to extract features more accurately by enhancing the quality of the fingerprint signal. This work proposes to use De-Convolutional Auto-Encoders for fingerprint image enhancement. Its performance is compared to seven state-of-the-art methods regarding their improvements for recognitions of the biometric system. Biometric performance is tested with MINDTCT and FingerJetFX for feature extraction and BOZORTH3 for biometric comparison. Critical comparisons are determined from 14 datasets. Those are used for evaluation of the methods. The impact of a method on biometric performance varies significantly. No single image enhancement can be found, which works best for all combinations. However, the proposed method *ConvEnhance* achieves highest count of best improvements among the evaluated methods.

## 5.1 Introduction

Using fingerprints for biometric recognition has a long tradition. Despite all advances and the maturity of fingerprint recognition technology, false positive and false negative errors still occur. One reason is the lack of quality of the signals processed.

Fingerprint recognition is about comparing two signals: the images captured from fingerprints. Usually, comparison of two fingerprints is not performed directly on the fingerprint images but on biometric features extracted from those images. A fingerprint recognition transaction therefore consists of two separate processes.[1] The first process is the *biometric feature extraction*, which extracts unique features from the biometric samples, e.g. the fingerprint minutiae from the fingerprint images. The second process is the *biometric comparison* which compares the biometric features of two biometric samples. The original input signal significantly influences the performance of the processing. Therefore, all three aspects are relevant for a successful biometric recognition: the quality of the fingerprint image itself, the biometric feature extraction, and the comparison algorithm to be applied.

Image enhancement can improve the aspect of the quality of the fingerprint image.

[1] The entire process starts with the acquisition of the fingerprints, which in itself is subject of numerous environmental factors and depends on human interaction. However, it is out of scope for this work.

This approach learns characteristic structures directly from the data. The structure of the data is later used to reconstruct the relevant signal even from low quality input. By doing so, this data-driven approach has the potential to outperform completely engineered approaches which may not be suited for all datasets.

It compares this approach to previous presented methods for enhancement in terms of their impact on biometric recognition performance. This impact will be evaluated by the accuracy achieved with publicly available feature extractors MINDTCT [309] and FingerJetFX [65] in combination with the comparator BOZORTH3 [309]. Critical comparisons are identified on fourteen datasets in order to illustrate the effect.

The rest of the publication is organized as follows: A selection of image enhancements is briefly explained in Section 5.2. Our approach will be explained in Section 5.3. Section 5.4 introduces the test protocol including datasets, metrics for testing, and the corresponding results. Conclusions are made in Section 5.5. Section 5.6 adds remarks on the findings of this work and gives an outlook on future work.

## 5.2 Related Work

Over the last decades several ideas were proposed on how to enhance fingerprint images. Seven methods will be described by how they manipulate the signals of fingerprints, i.e. the grey-values. Some methods are composed from several single processing steps. Where reasonable, single working steps are described and evaluated stand-alone.

*Histogram Equalization* (HistEQ) is a general method for image enhancement. Its idea of HistEQ is to remap the pixel intensities in such a way that the intensity distribution spreads over the resulting histogram as uniform as possible while keeping the order of the intensity values. By such mapping, the values of the signal are spread over the entire valid range and the contrast is stretched in a global manner. While HistEQ is applied to an entire image, *Contrast Limited Adaptive Histogram Equalization* (CLAHE) performs a contrast stretching of the signal in a more local manner [336]. The local adaptivity overcomes limitations of the simple HistEQ, e.g. it is able to enhance even local low-contrast areas. Unfortunately, CLAHE is not sensitive, whether or not a relevant signal in present locally. Therefore, the contrast of every local signal - including noise - will be amplified. For both HistEQ and CLAHE functions included in openCV have been used [32].

Hong et al. proposed another reshaping of the signal distribution of a fingerprint image [110]. [2] The intention of shaping the distribution of the signal is to let the mean and the standard deviation of the enhanced image meet distinct target values. By doing so, distributions of grey-values get more similar over a set of images. All resulting signals should have therefore similar energy.

Watson et al. assumed that locally the signal of a fingerprint is similar to an oriented 2D-cosine [305]. If so, this signal should be observable well in the frequency domain. Its specific frequency can then be enhanced more easily in the frequency domain than in the spatial domain. Thus, they proposed an enhancement in the frequency domain using a *Block-wise Fourier Spectrum Enhancement* (BlockFFT). It is proposed to amplify the spectrum of local image blocks according to its corresponding power spectrum. Assuming that the fingerprint ridge frequency itself is the dominant contributor in the spectrum this signal will be boosted the most. Processing of overlapping blocks is proposed to prevent artefacts at the borders of local blocks.

Greenberg et al. proposed to use Wiener Filters for fingerprint image enhancements [97].[3] The idea is to estimate the characteristics of noise and to remove such noise from signal.

Hong et al. proposed a method [110], which has later been called *Contextual Filtering* by Maltoni et al.. They state that it may be *"the most widely used technique for fingerprint image enhancement"* [203]. In contrast to most other approaches, contextual filtering is especially tuned to fingerprint image enhancement and focusses on its specific characteristics. Based on estimations for local orientation and ridge frequency appropriately parametrized Gabor filters are applied locally. With the usage of Gabor-Filters again the underlying assumption is, that the signal of a fingerprint is locally similar to an oriented cosine. Several extensions to Contextual filtering were proposed which modify one or more working steps, e.g. using the frequency domain for estimation of local orientation and ridge distance.

Buades et al. proposed a decomposition of the signal of the fingerprint image into two components [35]. Local Total Variation is used as a measure for sensibility regarding smoothing operations. This measure can be used to determine the so-called cartoon component which represents the very low frequency component which is less sensible to smoothing. The texture part shall contain a clearly visible

[2] This method is a single processing step in a larger enhancement workflow named Contextual Filtering

[3] Wiener Filtering is proposed as a first processing step in a larger enhancement workflow. It will be evaluated only stand-alone in this work. For this work an unsupervised variant of the Wiener filtering provided by `opencv` [32] is used.

fingerprint signal. The method is therefore called *Cartoon-Texture-Decomposition* (Cartoon).

While these seven evaluated image enhancement methods cover the methods from the *Handbook of Fingerprint Recognition* [203], more ideas have been proposed. A broad survey on methods is e.g. given by Ezhilmaran and Adhiyaman [74].

Sahasrabudhe et Namboodiri proposed an approach which is the most relevant for our own approach [251]. They propose to train a Convolutional Restricted Boltzman Machine (CRBM) with two hidden layers on hand-selected fingerprint image. Finally, they apply the train CRBM iteratively to reconstruct an enhanced image from a given fingerprint image. This approach is out of scope for the evaluations of this work.

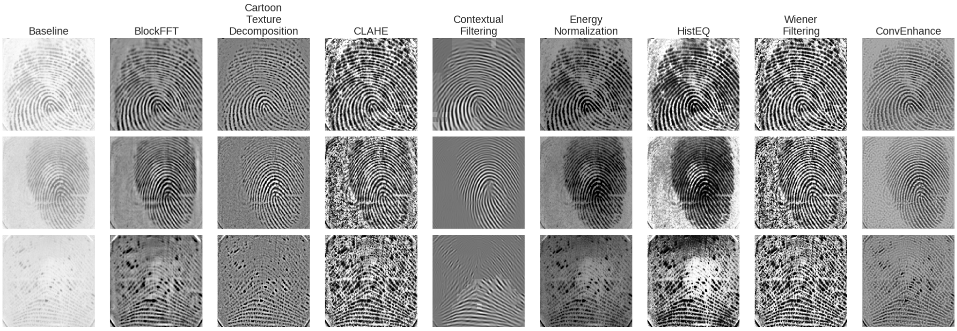


**Figure 5.1:** All evaluated enhancements applied to sample fingerprint images.

## 5.3 Proposed Method

*Deep Learning* and *Convolutional Neural Networks* (CNNs) have led to significant advances in the domains of image processing and pattern recognition [173]. These advances are promising for many pattern recognition tasks. Thus, we are motivated to explore its benefit for fingerprint recognition.

The central idea in CNNs is that digital filters are used as feature extractors. These feature extractors can be *trained* by systematic modification of the filter coefficients: the digital filters will *learn* to detect and extract features. The partial derivatives of the filter coefficients with respect to a given objective function can be used for a gradient descent and therefore induce such systematic modifications. The filters therefore learn features that help solving the given task.

The central idea of Deep Learning is to stack a multitude of sets (layers) of such trainable digital filters. An distinct set of combined layers is called a *model* $\mathcal{M}$. In the first layer of a CNN only very simple features will be learned, e.g. edges. The second layer learns re-combinations of the features extracted in the first layer. With each layer of non-linear re-combinations the extracted features can represent more complex structures. The capacity of a model for features that can be described grows with the number of layers and so does the complexity of describable features grow.

CNNs and techniques from Deep Learning are very versatile in their application. Thus, they can be used as a method for image enhancement of fingerprints. As CNNs combine local and more global features, they are promising for fingerprint image enhancements. In areas, where the fingerprint signal is of good quality, CNNs can rely more on the local data. For regions of low quality more global information (e.g. regional orientation) will be taken into account for local reconstruction. As CNNs learn directly from the data, CNNs automatically learn how to combine local and more global features. Especially De-ConvLayers have high potential of solving the problems in fingerprint image enhancement [327].

### 5.3.1 Architecture

Five types of layers were used during the experiments. Those were Convolutional, De-Convolutional, Rectified-Linear transformation, DropOut, and Normalization. A *Convolutional Layer* (ConvLayer) consists of trainable filters [172] and implements the ideas explained roughly above. A special type of the ConvLayer is the De-Convolutional Layer (De-ConvLayer) introduced by Zeiler et al. [327]. They can be used to reassemble local structures from features. The De-ConvLayers are therefore the most relevant layers for the reconstruction in this architecture since they can reconstruct larger regions from local representations in the layers before. The layout of ConvLayers and DeConvLayers can be described by the number of filters and dimensions of the filters. All filters are 3D-filters. According to their spatial behaviour, dimensions are therefore `HEIGHTxWIDTHxCHANNELS`[4]. A *Rectified Linear Unit* (ReLU) is a simple non-linear transformation, which sets all negative values to 0 and use the identity for all other values.[5] Despite its simplicity, ReLUs are helpful mainly for two reasons. They introduce sparsity in the output values [217] and they introduce additional non-linearities which increases the capacity of a model. *DropOut* randomly sets features to zero and serves to counteract over-fitting [272]. Finally, *Mean Variance Normalization* (MVN) layers will conduct a normalization on the input values. This leads to faster convergence and more accurate results. In general, all input values will be transformed such that after enhancement all values have zero mean and unit variance.

[4] Each filter generates its own feature channel. The number of channels is therefore equal to the number of filters of the predecessor layer.

[5] $\mathrm{ReLU}(x) = \max(0, x)$

**Table 5.1:** The evaluated model is composed from an ordered sequence of layers with distinct types. ConvLayer and De-ConvLayer can be described by their number of filters and the dimensions of those 3D-filters (`HEIGHTxWIDTHxCHANNELS`). MVN, ReLU, and DropOut layers do not have filters. The output of each layer is exemplary given with respect to an input image used for training.

| Nr | Type | Number of Filters | Dimensions of Filters | Output Dimensions |
|---|---|---|---|---|
| 0 | Input image | | | 336x256x1 |
| 1 | MVN | | | 336x256x1 |
| 2 | ConvLayer | 64 | 5x5x1 | 336x256x64 |
| 3 | MVN | | | 336x256x64 |
| 4 | ReLU | | | 336x256x64 |
| 5 | DropOut | | | 336x256x64 |
| 6 | De-ConvLayer | 64 | 5x5x64 | 336x256x64 |
| 7 | ReLU | | | 336x256x64 |
| 8 | DropOut | | | 336x256x64 |
| 9 | De-ConvLayer | 64 | 5x5x64 | 336x256x64 |
| 10 | ReLU | | | 336x256x64 |
| 11 | De-ConvLayer | 1 | 7x7x64 | 336x256x1 |

The De-ConvLayers used for reconstruction and the straight forward reconstruction are the central differences to the iterative CRBM approach of Sahasrabudhe et Namboodiri [251]. The model used in this work was designed and trained in the framework `caffe` [141]. The proposed model consists of eleven layers. The combination of layers can be found in Table 5.1. Among several evaluated layouts this architecture achieved the lowest reconstruction error for the training data. The enhancement of a single fingerprint image with dimensions 300x300 with this method takes about 75 ms on a GPU[6].

### 5.3.2 Training

Training of a model $\mathcal{M}$ to map an image $I$ to a target $T$ can be formulated as a quadratic optimization, i.e.

$$\min_{\mathcal{M}} ||T(I) - \mathcal{M}(I)||^2 \tag{5.1}$$

We propose the target to be an ideal binarization of the image. We use the synthetic fingerprint image generator `SFinGe` [46] to generate 50,000 synthetic fingerprint samples and corresponding ideal binarized images (see Figure 5.2 for samples).

[6] An NVIDIA GTX 780 has been used for evaluation.

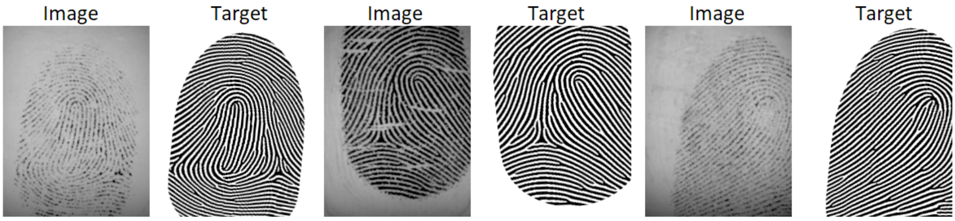


**Figure 5.2:** Three sample sets of images and targets used for training

These binarized versions $B(I)$ have been used as targets for training:

$$\min_{\mathcal{M}} ||B(I) - \mathcal{M}(I + \mathcal{N}_{0,\sigma})||^2 \tag{5.2}$$

where $\mathcal{N}_{0,\sigma}$ is white Gaussian noise, which acts like a corruption of the input image. This corruption can increase the capability of generalization of a model $\mathcal{M}$ [298]. This method tries to approximate a binarization of a local signal with respect to its neighbourhood.

The training has been performed with a learning rate $\lambda = 5 * 10^{-8}$. Errors have been tracked on an 800 sample validation set of synthetic images for early stopping. Weight decay has been set to $10^5$.

As training is carried out on synthetic data which is totally separated from the data used for testing, there is no need for techniques like cross-validation.

## 5.4 Experiments

### 5.4.1 Data

Accuracy of the biometric recognition needs to be evaluated on representative data. This has to be done in such a manner that results are reproducible. Evaluations in this work were performed on fourteen publicly available datasets. The benchmark series *Fingerprint Verification Competition* (FVC) has four editions providing datasets [200][201][202][47]. Five out of sixteen datasets from these series have been excluded from evaluation: one dataset with low resolution and all synthetic datasets. The *Ministerio de Ciencia y Tecnología* of Spain (MCYT) has provided a multi-modal data-set containing two subsets of fingerprint samples [228]: *MCYT DP* and *MCYT PB*. NIST provides the data-set *NIST SD14* [306]. Table 5.2 gives an overview over the evaluated datasets and their characteristics.

These three sets MCYT PB, MCYT DP, and NIST SD14 are in an order of magnitude larger than the other datasets. All evaluations on those datasets have been

**Table 5.2:** Datasets used for evaluation and Training. Characteristics are strong rotation (R), displacement (D), moisture (M), low quality (Q), and control on the acquisition (C).

| Dataset | Samples | Dimensions | Resolution | Data Origin | Characteristics |
|---|---|---|---|---|---|
| FVC2000 DB1 | 800 | 300×300 | 500 dpi | Optical | Low-cost Sensor |
| FVC2000 DB2 | 800 | 364×254 | 500 dpi | Capacitive | Low-cost Sensor |
| FVC2000 DB3 | 800 | 478×448 | 500 dpi | Optical | |
| FVC2002 DB1 | 800 | 388×374 | 500 dpi | Optical | R, D, M |
| FVC2002 DB2 | 800 | 560×296 | 569 dpi | Optical | R, D, M |
| FVC2002 DB3 | 800 | 300×300 | 500 dpi | Capacitive | R, D, M |
| FVC2004 DB1 | 800 | 640×480 | 500 dpi | Optical | R, D, M |
| FVC2004 DB2 | 800 | 364×328 | 500 dpi | Optical | R, D, M |
| FVC2004 DB3 | 800 | 480×300 | 512 dpi | Thermal Sweeping | R, D, M |
| FVC2006 DB2 | 1680 | 560×400 | 569 dpi | Optical | R, D, M, Q |
| FVC2006 DB3 | 1680 | 500×400 | 500 dpi | Thermal Sweeping | R, D, M, Q |
| MCYT DP | 39,600 | 400×256 | 500 dpi | Optical | C |
| MCYT PB | 39,600 | 300×300 | 500 dpi | Capacitive | C |
| NIST SD14 | 54,000 | 768×832 | 500 dpi | Ink-based | |
| Training Data | 50,000 | N×M | ∼ 500 dpi | SFinGe | Synthetic |

carried out only on first ten percent. The subsets can still be assumed representative for the evaluation since those subsets are still larger than the other sets.

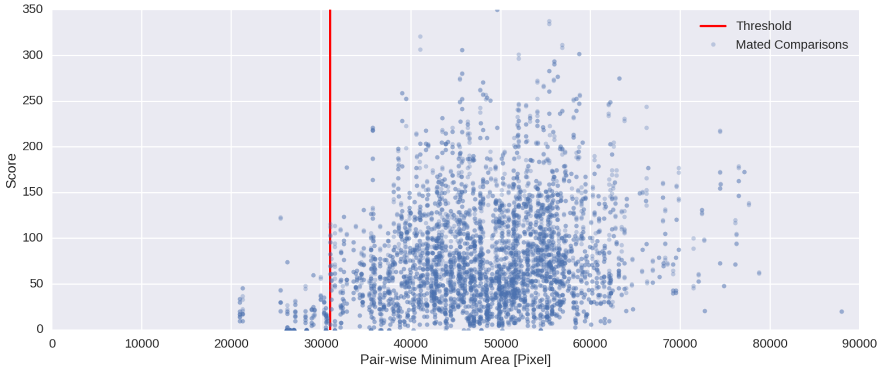


**Figure 5.3:** Low comparison scores for mated comparison can arise from low fingerprint area.

Specific subsets of image pairs have been selected systematically to focus evaluation on the most relevant cases of the biometric comparisons. The selection consists of two steps.

First, only those images $I$ will be kept where the area $A(I)$ of the fingerprint is large enough to provide sufficient information for comparison. The intention with this is to avoid genuine comparisons that are attributed with a low comparison score due to a small overlapping region of the image pair. The area $A$ is estimated by one of the features of NFIQ2.0 [222]. The subset $DB_{Large}$ of a whole dataset of fingerprints $DB$ is defined as follows:

$$\mathrm{DB}_{\mathrm{Large}} = \{I \in \mathrm{DB} : A(i) >= \mu_{A(\mathrm{DB})} - 2\sigma_{A(\mathrm{DB})}\} \tag{5.3}$$

where $\mu_{A(\mathrm{DB})}$ is the mean over all fingerprint areas in the entire dataset DB and $\sigma_{A(\mathrm{DB})}$ is the respective standard variation.

Figure 5.3 shows the comparison scores against the pair-wise minimum fingerprint area for all mated comparisons on the dataset FVC2000DB1. Small fingerprint areas may lead to small overlap areas between fingers. Image enhancement should not be able to increase the fingerprint areas. Therefore, the challenge of small overlap is none an image enhancement can take care of. Thus, those samples were not considered in our analysis.

Second, only those comparisons are selected which are most relevant for the biometric comparison performance. In general, the vast majority of biometric comparisons result in comparison scores, which allow to easily decide right whether it has been a mated or a non-mated comparison. Improvements on those comparisons are of little relevance for the biometric performance. Usually, the biometric performance depends on only a small subset of biometric comparisons in which the resulting comparison scores do not allow a sound decision if the comparison has been mated or non-mated. Those are the mated comparison resulting in very low comparison scores and vice versa for the non-mated comparisons. Only those can lead to false positive and false negative errors. As the focus of this work is on the influence of image enhancement on the biometric performance, we will reduce our analysis to the lowest scoring mated comparisons and highest scoring non-mated comparisons. For each dataset features will therefore be extracted from the non-enhanced original images with `MINDTCT` and `FingerJetFX`. Comparison will be carried out on both sets of extracted features. From all comparison results the 2,000 non-mated comparisons resulting in the highest biometric comparison scores and the 2,000 mated comparisons resulting in the lowest biometric comparison scores. Those 4,000 comparisons are of highest relevance for the biometric performance and form the fixed set of *critical comparisons* $C_{\mathrm{FE}}^{\mathrm{DB}}$. To inspect the impact of image enhancement those critical comparisons $C_{\mathrm{FE}}^{\mathrm{DB}}$ will be performed again on features from the enhanced images. To clarify this: these sets of critical comparisons depend only on the feature extractor FE but not on the image enhancement method Enh.

### 5.4.2 Metric

To assess the biometric performance all critical comparisons will be repeated with features extracted from the enhanced images. Figure 5.4 show the workflow ap-

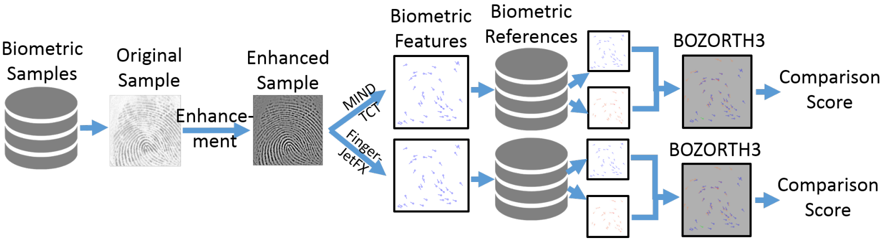


**Figure 5.4:** Workflow on the evaluation

plied to the critical comparisons. The fingerprint images involved in the critical comparisons $C_{\mathrm{FE}}^{\mathrm{DB}}$ are enhanced with an enhancement Enh. Features will be extracted from the enhanced images with MINDTCT and FingerJetFX. Comparison will be performed by BOZORTH3. The Area under the Curve AuC will be calculated for the resulting detection error tradeoff curve (DET) and used as the measure $m(\mathrm{Enh}, \mathrm{FE}, \mathrm{DB})$:

$$m(\mathrm{Enh}, \mathrm{FE}, \mathrm{DB}) = \mathrm{AUC}_{\mathrm{DET}}^{\mathrm{FE}}(\mathrm{Enh}, C_{\mathrm{FE}}^{\mathrm{DB}}) \tag{5.4}$$

The DET curve is evaluated in linear scale and the AuC is calculated for the entire range or errors by intention: as only the small set of critical mated and non-mated comparisons are inspected, logarithmic scaling would be misleading. Finally, values $m$ relative to the values before image enhancement are used as the relevant measure $m_{\mathrm{rel}}$:

$$m_{\mathrm{rel}}(\mathrm{Enh}, \mathrm{FE}, \mathrm{DB}) = \frac{m(\mathrm{Enh}, \mathrm{FE}, \mathrm{DB})}{m(\mathrm{Baseline}, \mathrm{FE}, \mathrm{DB})} \tag{5.5}$$

where ’Baseline’ indicates the identity function, i.e. the original image are evaluated.

Computational complexity is an important criterion for assessing image enhancement methods. The processing time for a fingerprint sample image of dimensions 300x300 will therefore be measured for all methods.

### 5.4.3 Results

Figures 5.5 and 5.6 visualize exemplary the DET curves for all evaluated methods of image enhancement for the critical comparisons $C_{\mathrm{MCYT_{DP}}}^{\mathrm{MINDTCT}}$ and respectively $C_{\mathrm{MCYT_{DP}}}^{\mathrm{FingerJetFX}}$. The solid black lines mark the baseline performance where no image enhancement has been applied. Such DET curves have been used to calculated the measures $m_{rel}$ in Table 5.4. Those measures $m_{\mathrm{rel}}$ are relative to the measures $m$ in Table 5.3. Since the baseline performances tend to be better for MINDTCT the relative improvements for this extractor have other characteristics than for FingerJetFX.

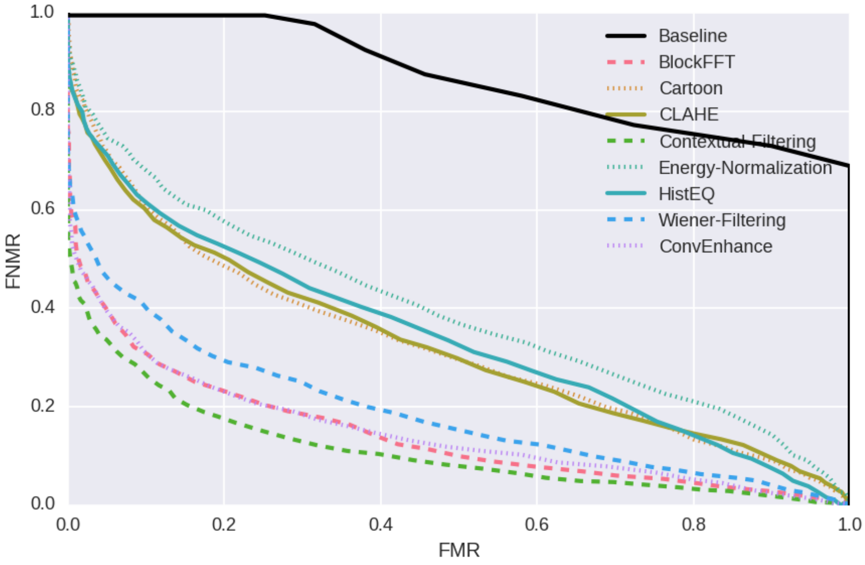


**Figure 5.5:** DETs on $C_{\text{MINDTCT}}^{\text{MCYT DP}}$

**Table 5.3:** Absolute values for measure $m(\text{Baseline}, \text{FE}, \text{DB})$ allow comparison of baseline performances between datasets and feature extractions.

| FE | DB | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT PB | MCYT DP | NIST SD14 |
| MINDTCT | 0.33 | 0.26 | 0.67 | 0.16 | 0.12 | 0.67 | 0.76 | 0.69 | 0.68 | 0.40 | 1.00 | 0.87 | 0.99 | 0.52 |
| FingerJetFX | 0.71 | 0.21 | 0.96 | 0.54 | 0.43 | 0.73 | 1.00 | 0.92 | 0.67 | 0.98 | 1.00 | 0.70 | 1.00 | 1.00 |

Figure 5.7 visualizes the distributions of the achieved relative measures $m_{\text{rel}}$ per dataset and feature extractor for all related methods as violin plots[7]. The left, blue halves of the violins represent the distribution of achieved relative AuC for all related image enhancements for the dataset denoted on the x-axis with features extracted with MINDTCT. The right, green halves represent the values for features extracted with FingerJetFX. Red stars indicate the improvement achieved by the proposed method ConvEnhance. The stars therefore give an idea about the performance of ConvEnhance with respect to the other methods of image enhance-

[7] These plots are slightly inaccurate since the number of measurements is small.

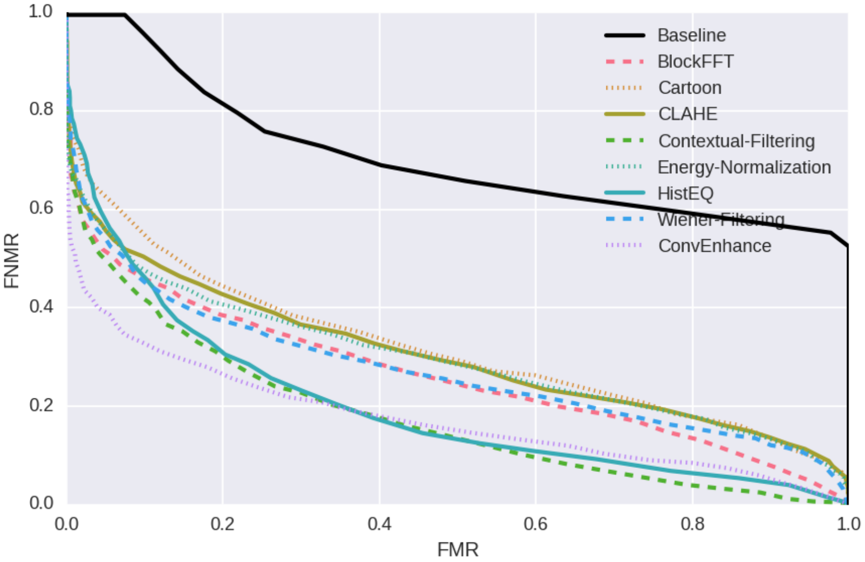


**Figure 5.6:** DETs on $C^{\text{MCYT DP}}_{\text{FingerJetFX}}$

**Table 5.4:** Measure $m_{\text{rel}}(\text{Enh}, \text{FE}, \text{DB})$. Smaller values indicate stronger improvements. The best improvement per dataset and feature extraction is marked bold.

| FE | Enh | DB | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2006 DB2 | FVC2006 DB3 | MCYT PB | MCYT DP | NIST SD14 |
| MINDTCT | BlockFFT | 0.53 | 0.36 | 0.46 | 0.59 | 1.40 | 0.40 | 0.64 | 0.56 | 0.59 | 0.43 | 0.72 | 0.16 | 0.70 | **0.40** |
| | Cartoon | 0.32 | 0.54 | 0.54 | **0.55** | 0.67 | 0.41 | 0.56 | 0.38 | 0.48 | 0.38 | 0.59 | 0.37 | 0.79 | 0.52 |
| | CLAHE | 0.36 | 0.37 | 0.55 | 0.62 | 0.70 | 0.46 | 0.63 | 0.42 | 0.63 | **0.15** | 0.81 | 0.38 | 0.75 | 0.57 |
| | Contextual-Filtering | 0.54 | 0.77 | 0.45 | 1.31 | 1.16 | 0.41 | 0.74 | 0.39 | 0.61 | 0.40 | **0.41** | **0.13** | **0.41** | 0.59 |
| | Energy-Normalization | 0.49 | 0.74 | 0.71 | 1.03 | 0.72 | 0.49 | 0.80 | 0.59 | 0.73 | 0.31 | 0.94 | 0.45 | 0.90 | 0.67 |
| | HistEQ | 0.46 | 0.41 | 0.58 | 0.68 | **0.66** | **0.40** | 0.67 | 0.45 | 0.64 | 0.22 | 0.83 | 0.39 | 0.76 | 0.60 |
| | Wiener-Filtering | 0.33 | 0.31 | 0.55 | 0.63 | 0.70 | 0.45 | 0.61 | **0.37** | 0.56 | 0.19 | 0.69 | 0.22 | 0.59 | 0.61 |
| | *ConvEnhance* | **0.29** | **0.28** | **0.32** | 0.65 | 1.14 | 0.40 | **0.50** | 0.46 | **0.44** | 0.17 | 0.50 | 0.17 | 0.47 | 0.71 |
| FingerJetFX | BlockFFT | 0.21 | 0.61 | 0.57 | 0.89 | 0.63 | 0.57 | 0.67 | 0.51 | 0.63 | 0.51 | 0.86 | 0.37 | 0.69 | 0.89 |
| | Cartoon | 0.11 | **0.55** | 0.61 | 0.78 | 0.58 | 0.47 | 0.66 | 0.46 | 0.68 | 0.47 | 0.97 | 0.45 | 0.77 | 0.87 |
| | CLAHE | 0.14 | 0.65 | 0.54 | 0.70 | 0.64 | 0.40 | 0.61 | 0.34 | **0.56** | **0.29** | 0.80 | 0.43 | 0.65 | 0.79 |
| | Contextual-Filtering | 0.22 | 1.61 | 0.59 | 1.02 | 0.78 | 0.81 | 0.70 | 0.50 | 0.80 | 0.56 | 0.76 | 0.25 | 0.76 | 0.89 |
| | Energy-Normalization | 0.13 | 0.66 | 0.62 | 0.76 | 0.60 | 0.49 | 0.67 | 0.53 | 0.74 | 0.52 | 0.96 | 0.42 | 0.83 | 0.89 |
| | HistEQ | 0.17 | 0.63 | 0.57 | 0.66 | 0.58 | 0.45 | **0.53** | 0.43 | 0.66 | 0.48 | 0.83 | 0.28 | 0.71 | 0.80 |
| | Wiener-Filtering | 0.14 | 0.57 | **0.52** | **0.64** | 0.63 | **0.39** | 0.57 | 0.34 | 0.56 | 0.31 | **0.74** | 0.39 | 0.61 | **0.79** |
| | *ConvEnhance* | **0.08** | 0.56 | 0.60 | 0.76 | **0.49** | 0.46 | 0.65 | **0.33** | 0.65 | 0.38 | 0.84 | **0.25** | **0.52** | 0.87 |

ment. In most cases ConvEnhance achieves a performance better than the mean of all other methods.

Improvements vary significantly for the different datasets and feature extractions. There is no single best image enhancement that provides the best improvement for all combinations of datasets and feature extractions. An enhancement working well for one feature extraction on a single dataset in some cases does not work at all with another combination. There is no obvious trend in performance in dependence to origin of data[8].

The results for datasets FCV2002 DB1 and DB2 and features extracted with MINDTCT have to be approached with caution. The AuC values are already low for the baseline. Thus, only very few comparisons can have an erroneous outcome at all. The changes in the results depend on a small number of comparison what may make the results sensitive to noise. For those two datasets $m_{rel}$ values larger than 1 can be found, i.e. the biometric performance even decreased after an image enhancement.

For features extracted with MINDTCT the most frequent best improvement is achieved with ConvEnhance (5 best improvements). The next most best improvements are achieved by Contextual-Filtering (3) and HistEQ (2). The other best results are achieved by the other methods.

ConvEnhance and Wiener Filtering both achieved the most best improvements (5 each) for features extracted with FingerJetFX. This might be a an indication that FingerJetFX is sensitive to noise. The next best improvements are achieved by CLAHE (2). The remaining best results again spread over different methods of enhancement.

It is remarkable that in some constellations very simple image enhancements like HistEQ outperform the more sophisticated approaches.

The computational complexity of an image enhancement can be an important criterion when designing a system. Table 5.5 summarizes the processing time when the image enhancement methods are applied to a fingerprint sample image with dimensions 300x300. No runtime optimized versions of the image enhancement methods have been tested. Therefore, the timing measurements can only be taken as a very rough indicator for the computational complexity. While HistEQ and CLAHE process an image in less than a millisecond, the processing time for Contextual Filtering and ConvEnhance (when working on CPU only) is larger by orders of magnitude. However, as CNNs are suited for operation on GPU, ConvEnhance can be processed in about $75\,\mathrm{ms}$ when using GPU.

[8] See Table 5.2 for the origin of the data for each dataset.

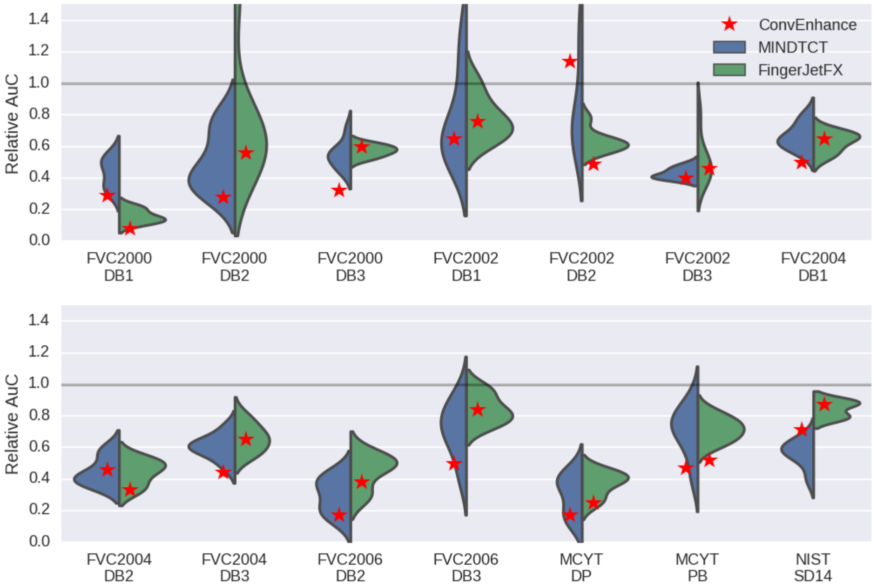


**Figure 5.7:** Violin plot for AuC. Red stars mark the values for ConvEnhance

**Table 5.5:** The evaluated methods differ is their numerical effort by orders of magnitude. The processing time is measured for the implementations applied to a 300x300 fingerprint sample image.

| Method | Processing Time [ms] |
|---|---|
| BlockFFT | 74 |
| Cartoon Texture Decomposition | 18 |
| CLAHE | $< 1$ |
| Contextual Filtering | 13,599 |
| ConvEnhance (GPU) | 75 |
| ConvEnhance (CPU) | 3,010 |
| Energy Normalization | 3 |
| HistEQ | $< 1$ |
| Wiener Filtering | 443 |

## 5.5 Conclusion

Several methods have been proposed for the enhancement of fingerprint images. Seven related methods and a newly proposed one have been evaluated for their impact on biometric performance on fourteen datasets and two different feature extractors. No single best image enhancement can be found and the impact of enhancement strongly varies. The best improvement depends on both: data and feature extraction. The proposed method ConvEnhance performs competitive against the other image enhancements. It achieves the highest count of best improvements over all combinations of datasets and feature extractors. Given this measure it outperforms all other methods.

The computational time is about 75ms for the enhancement of an image with dimension 300x300. This amount of time seems reasonable for operational use but a GPU is necessary.

## 5.6 Discussion and Outloook

Optimization of hyper-parameters like the layout of single layers is essential in Deep Learning. We do not claim that the proposed architecture is the best but it can be used for further improvements. Especially, if a cost function can be formulated which is more related to the biometric performance, a more powerful image enhancement may be learned.

Synthetic fingerprint images were used for training. The model has learned how such fingerprint images have to be enhanced. This approach may fail, if the fingerprint data which is to be enhanced differs significantly in its characteristics from training data, e.g. in its image resolution. Changing the data source to authentic fingerprint images may increase the performance of a trained model $\mathcal{M}$ or to apply to specific characteristics of the data of a target system.

Fingerprint image quality can be also be assessed by specialized metric NFIQ2.0. This metric therefore would be an alleged candidate for the assessment of the impact of image enhancement on fingerprint recognition.

Using the AuC on only a subset of all data is always a simplification. This was done for a better illustration the effect of image enhancement on biometric performance. For application, one shall always consider the performance on the entire datasets.

It was found that the method of image enhancement, which is suited best, depends on the feature extractor. Further methods of image enhancement might be considered for comparison. Combinations of image enhancements as cascades are conceivable. Such cascades may be an appropriate approach to deal with challenges specific for the data or feature extractor used. More feature extractors and more comparison algorithms might be considered for evaluation.

# Chapter 6

# Minutia-based Enhancement of Fingerprint Samples

**Summary** Fingerprint feature extraction shall be as accurate and reliable as possible. An ideal fingerprint sample enhancement takes these aspects into account. We introduce a cost function for a Convolutional Neural Networks, which incorporates the accuracy and reliability of the feature extraction. This allows to tailor an image enhancement directly to the needs of a feature extraction module. Evaluations of the biometric performance demonstrate the potential of this approach. This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was presented at the *International Carnahan Conference on Security Technology* in Madrid (Spain) in 2017.

[258] P. Schuch, S. Schulz, and C. Busch. Minutia-based enhancement of fingerprint samples. In *International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2017

**Abstract** Image enhancement is a common pre-processing step before the extraction of biometric features from a fingerprint sample. This can be essential especially for images of low image quality. An ideal fingerprint image enhancement should intend to improve the end-to-end biometric performance, i.e. the performance achieved on biometric features extracted from enhanced fingerprint samples. We use a model from Deep Learning for the task of image enhancement. This work's main contribution is a dedicated cost function, which is optimized during training. The cost function takes into account the biometric feature extraction. Our approach intends to improve the accuracy and reliability of the biometric feature extraction process: No feature should be missed and all features should be extracted as precise as possible. By doing so, the loss function forced the image enhancement to learn how to improve the suitability of a fingerprint sample for a biometric comparison process. The effectivity of the cost function was demonstrated for two different biometric feature extraction algorithms.

## 6.1 Introduction and Motivation

Fingerprint recognition requires careful selection of suitable signal processing algorithms. This signal processing is about comparing two signals for their similarity. The signals are fingerprint samples that represent friction ridges. In general, the comparison is not performed on the fingerprint samples directly. Usually, distinct biometric features are extracted from the fingerprint samples, which can be evaluated to a comparison subsystem. Typical examples of such features are the fingerprint minutiae, e.g. the endings and bifurcations of the fingerprint ridges. The higher the quality of a fingerprint sample is, the better the biometric features can be extracted. But feature extraction can be challenging when low quality samples are processed. The accuracy of fingerprint recognition therefore depends remarkably on the quality of the fingerprint samples.

The quality of the fingerprint samples varies. Good quality fingerprints can usually be processed accurately. But the lower the quality is, the more challenging is the biometric feature extraction. Fingerprint image enhancement (FIE) can improve the suitability for accurate feature extraction of the fingerprint sample. FIE is a common preprocessing step before the biometric feature extraction process [203]. Figure 6.1 visualizes the typical workflow of biometric fingerprint recognition including a FIE.

The intention of the FIE should be to improve the performance. The enhancement therefore does not need to improve the aesthetics of a fingerprint sample. It should intent to increase the utility of a fingerprint sample for biometric comparison [123]. Other proposed approaches for FIE usually assess the improvement of biometric performance after enhancement. But those approaches cannot address the performance directly. Actually, this is the intended purposes at the end. But this also is a very indirect approach as it also incorporates the comparison. The biometric feature extraction is the successive processing step after image enhancement. It is therefore reasonable to enhance the samples in such a way, that the feature extraction algorithm can extract the minutiae as accurate and reliable as possible. Accuracy means to extract features with a high fidelity in position. Reliability covers two partial aspects. No spurious minutia should be found and no actually present minutia shall be missed out.

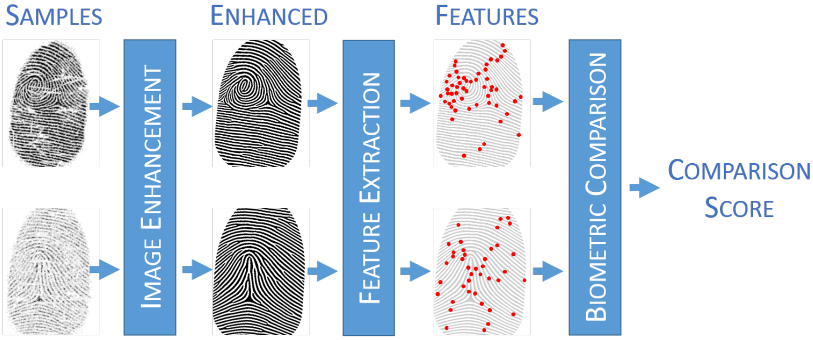


**Figure 6.1:** The workflow of fingerprint recognition can roughly be described in three sequential processes. First the fingerprint samples are enhanced. Biometric features are extracted afterwards. Finally, the features of two samples are compared to estimate the likelihood of being samples from the same finger.

Deep Learning (DL) is quite successful in many tasks of image processing nowadays. One of the important assets of DL is the ability to learn how to solve a task directly from training data. It is therefore also promising for FIE.

Thus, we propose to train a DL model for FIE. We chose a convolutional Auto-Encoder for image enhancement and sketched an exemplary model architecture. This work's main contribution is our proposed cost function. The cost function is optimized during training. Formulation of the cost function usually is a fundamental design step, because it defines what a model will actually learn during training. Our cost function addresses the aspects of accuracy and reliability of a feature extraction. It compared minutiae extracted from the feature extractor against ground truth minutiae. As different feature extractors work differently, the image enhancement can be adjusted to the needs of the feature extractor in this way. We demonstrated this approach for two different feature extractors: `MINDTCT` and `FingerJetFX`.

The rest of this work is organized as follows: Section 6.2 contains an overview on related work. Our approach is presented in section 6.3 and is evaluated in Section 6.4. Section 6.5 concludes the findings of this work.

## 6.2 Related Work

A lot of methods for FIE were proposed before. Several surveys on FIE allow an overview over general methods in FIE methods [74][209][208][146][2][254][238].

Most relevant for our approach were those methods, which used DL or representations in general for FIE. There are DL approaches which train a reconstruction of ideal ridge structures from fingerprint samples [257][276]. Ideal ridge structure were provided by synthetic fingerprint generation tool SFinGe [46]. Deep Belief Network built from Boltzman Machines were proposed for FIE [251]. Auto-Encoders were used to learn representations for minutia and non-minutia patches [253]. Those representations were later used to classify the local presence of minutia which in turn should enhance biometric performance. There are also indirect approaches, which use a DL based orientation field estimation for FIE [250][133].

Learning dictionaries for fingerprint image enhancement has the idea in common with DL, that it is possible to learn representations directly from the data, which enable an image enhancement. There are approaches, which use dictionaries of ridge structure patches directly for image enhancement [154][43][133][270][167][190][300][190][311]. Singh et al. proposed dictionaries for super-resolution [269]. Super resolution can also be understood as an enhancement or reconstruction. Cao et Jain reconstructed fingerprints from fingerprint minutia via dictionaries [39]. This is kind of an inverse problem to ours, but it is quite similar as the aim was to reconstruct fingerprint samples which would end up in the same minutiae after feature extraction. Like for the DL approaches, there are also indirect approaches, which learn an orientation field for later FIE [321][322][77][319]. Finally, there are also hybrid approaches, which combine dictionaries and techniques from DL [136][40].

Another approach for FIE based on a decomposition into representations was to use Morphological Component Analysis [57].

All approaches assessed the impact of FIE on the biometric performance. But none is able to incorporate biometric feature extraction or biometric comparison directly.

Our proposed approach uses a typical Auto-Encoder training in the first training phase. The main contribution of this work is the cost function used during the second training phase. The cost function exceeds all the related work by allowing to incorporate the accuracy and reliability of a biometric feature extraction directly into the training of the method.

## 6.3 Our Approach

We use a convolutional Auto-Encoder. In the following, we sketch an exemplary architecture first. The actual architecture of the model is not that important for this work at the end of the day, since our focus was on the cost function. Afterwards, the two successive training phases are explained. The second step includes our main contribution: the designated cost function. Finally, we describe the signal normalization, which was applied to the fingerprint samples before enhancement. Such a normalization is usually helpful when a model is applied to data that is unseen during training.

### 6.3.1 Architecture

We used DL framework `caffe` for the design and training of the model $\mathcal{M}$ [141]. We decided to train a Convolutional Auto-Encoder because the main task of an FIE should be to recover the relevant structure from an input. The model $\mathcal{M}$ consists of Convolutional layers (ConvLayer), DeConvolutional Layers (DeConvLayer), a MaxPooling layer and two type of activation layers (ReLU and Sigmoid).

The model's architecture is visualized in Figure 6.2. The architecture was composed of three parts. One part is responsible for local representations. It is composed of three consecutive blocks of ConvLayers and ReLU. Each ConvLayer operates with a striding of 2. This results in an eight-fold subsampling, which is about half the distance between two fingerprint ridges when processing images of 500dpi resolution. The second part of the structure should take care of regional representations. Therefore, this part works partially at 32-fold subsampling. This means, that a single pixel represents a region with the width of about two fingerprint ridges in this representation. At the end of the block of regional representation, the information is upscaled by a DeConvLayer. The third and last block reconstructs the image from the local and the regional representations. This is done by a DeConvLayer and a ConvLayer, which works like a proxy for a fully connected layer.

### 6.3.2 Auto-Encoder Pre-Training

In the first phase, the model $\mathcal{M}$ was trained unsupervised in a *de-noising* manner [298]. This pre-training allowed to learn a reasonable representation for fingerprint samples. Let $I$ be the input image and let $\mathcal{N}_{0,\sigma}$ be Gaussian Noise with zero mean

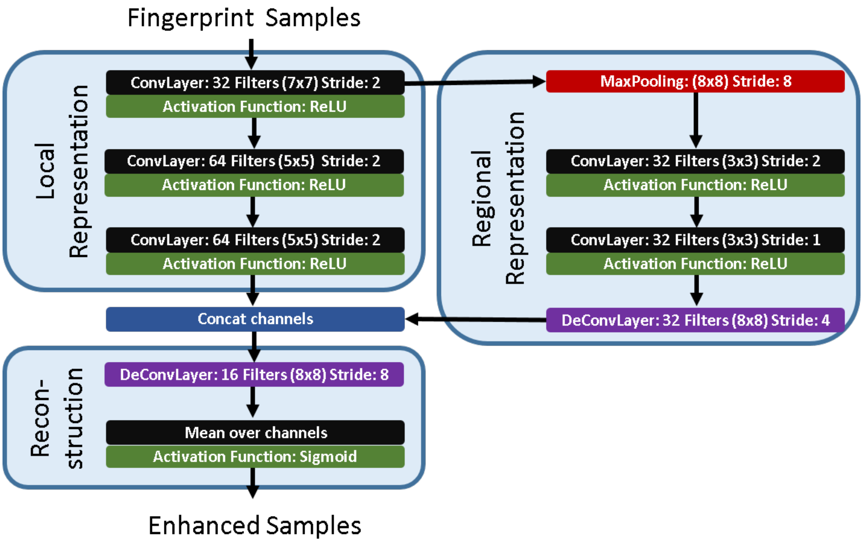

Fingerprint Samples
Local Representation
ConvLayer: 32 Filters (7x7) Stride: 2
Activation Function: ReLU
ConvLayer: 64 Filters (5x5) Stride: 2
Activation Function: ReLU
ConvLayer: 64 Filters (5x5) Stride: 2
Activation Function: ReLU
Regional Representation
MaxPooling: (8x8) Stride: 8
ConvLayer: 32 Filters (3x3) Stride: 2
Activation Function: ReLU
ConvLayer: 32 Filters (3x3) Stride: 1
Activation Function: ReLU
DeConvLayer: 32 Filters (8x8) Stride: 4
Concat channels
Recon-struction
DeConvLayer: 16 Filters (8x8) Stride: 8
Mean over channels
Activation Function: Sigmoid
Enhanced Samples

**Figure 6.2:** The model $\mathcal{M}$ consists of ConvLayers, De-ConvLayer, MaxPooling layers, and activation layers.

and standard deviation $\sigma$. The cost function is then defined as follows:

$$\min_{\mathcal{M}} ||I - \mathcal{M}(I + \mathcal{N}_{0,\sigma})||^2 \tag{6.1}$$

The model $\mathcal{M}$ learns to reconstruct the image $I$ from the corrupted version $I + \mathcal{N}_{0,\sigma}$. The model was trained by optimizer *Stochastic Gradient Descent*. The batch size in each training step was 64 of randomly sampled training images. Each sample was a random crop of size $256 \times 256$ pixels from the training images. Learning rate was initialized with $10^{-4}$. It was adjusted during training according to the *Inverse* policy ($\gamma = 10^{-4}$ and power $= 0.75$). A small weight decay coefficient of $5 \cdot 10^{-8}$ was used to prevent overfitting. Momentum was set to $0.5$. Additive Gaussian Noise was generated with a standard deviation $\sigma = 0.03$.

### 6.3.3 Minutia-based Fine-Tuning

The pre-trained model was then fine tuned. We therefore exchanged the Auto-Encoder cost function by our minutia-based cost function which is described in the following.

A derivable cost function needed to be established to allow training of the model. The cost function should take the accuracy and reliability of the biometric features into account. Let us first focus on the accuracy. In order to achieve higher accuracy, the cost function needs to relate the positions of the ground truth minutiae to the extracted ones in a derivable manner.[1] This approach therefore assumes knowledge of the ground truth minutiae. We introduced such relation information by placing Gaussians at the positions of minutiae. This gave rise to a new reasonable 2D representation for a set on minutiae. Figure 6.3 visualizes such a representations over the positions of minutiae. This 2D representation can then be used for formulation of the cost function.

The cost function was defined as the mean squared error of all pixels $(x, y)$ between the target representation $\vartheta|_{(x,y)}$ for the ground truth minutiae and the actual representation $\alpha(g)|_{(x,y)}$ for the extracted minutiae. The actual $\alpha(g)|_{(x,y)}$ representation depends on the local grey value $g$. The cost function is then defined as sum over the squared distance between both representations at every pixel:

$$cost = \sum^{x} \sum^{y} (\vartheta|_{(x,y)} - \alpha(g)|_{(x,y)})^2 \tag{6.2}$$

For improved readability, the local dependency is left out in the following. The derivation of this cost function with respect to the grey value $g$ is as follows:

$$\begin{aligned}
\frac{\mathrm{d}cost}{\mathrm{d}g} =& \frac{\mathrm{d}(\vartheta - \alpha(g))^2}{\mathrm{d}g} \\
=& 2(\vartheta - \alpha(g)) \frac{\mathrm{d}(\vartheta - \alpha(g))}{\mathrm{d}g} \\
=& 2(\vartheta - \alpha(g)) \lim_{\epsilon \to 0} \frac{\left( (\vartheta - \underbrace{\alpha(g + \epsilon)}_{:= \alpha^+}) - (\vartheta - \alpha(g)) \right)}{\epsilon} \\
=& 2(\vartheta - \alpha(g)) \lim_{\epsilon \to 0} \frac{\alpha - \alpha^+}{\epsilon}
\end{aligned} \tag{6.3}$$

The derivative of the cost function is built by chain rule. The inner function $(\theta - \alpha(g))$ cannot not be differentiated analytically. Thus a numerical approximation using $\alpha^+ := \alpha(g + \epsilon)$ is used. This gradient could then be used during training the model.

Minimizing the cost function forces the model to generate a representation $\alpha$ as similar as possible to representation $\vartheta$. It therefore takes the accuracy into account, because this forces the Gaussians in representation $\alpha$ to be placed similar to those in representation $\vartheta$. The cost function also considers the reliability of the feature

[1] There are more features of a minutia beside its position. Features like minutia type and angle are out of scope for this work.

extraction, since missed out features or spurious features result in large distances between representations $\vartheta$ ans $\alpha$.

The pre-training was assumed to learn a reasonable representation of fingerprints in the layers, which were responsible for the local and the regional representations. As pre-training was finished, we froze the training for those layers, i.e. the parameters were not altered by the training anymore. Only those layers were fine tuned, which took care of the final reconstruction. The model was trained with similar training parameters as during the pre-training. Only the learning rate was set to $0.05$ and no regularization by weight decay was enforced.

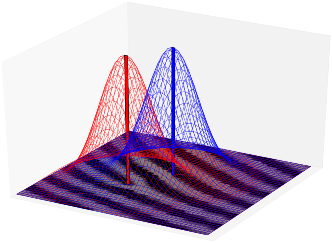


**Figure 6.3:** Extracted minutia positions (blue bar) and ground truth (red bar) should be as close as possible. Placing Gaussians at the positions of the minutiae induced a new representation. The target representation $\vartheta$ (red Gaussian) and actual representation $\alpha$ (blue Gaussian) should be as similar as possible. The difference between both representations can be used to establish local gradients which are necessary for training the model.

### 6.3.4 Data Normalization

Each fingerprint image was contrast enhanced, before DL based enhancement was applied. Even though such a normalization could be trained within the model, a normalization is a common pre-processing step. This normalization aims to reduce differences in signal characteristics between different image. The simplifies the training process. We chose a simple contrast stretching, which is robust to statistical outliers. The highest and lowest $3\%$ of all grey values in each images were clipped and the remaining grey values were stretched to the range $[0, 1]$.

## 6.4 Results

### 6.4.1 Data

Using the proposed cost function required ground truth minutia positions for the training images. We used the tool *SFinGe* for generation of such a dataset. SFinGe provided such information for each generated fingerprint [46]. We used the tool's

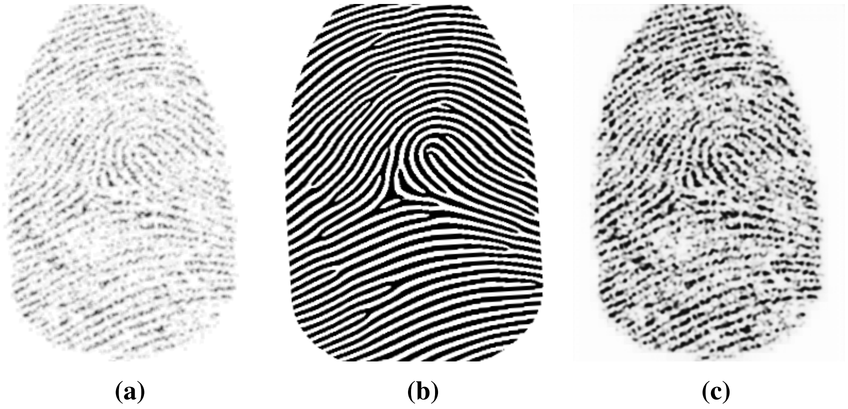


**Figure 6.4:** A fingerprint sample (6.4(a)) usually contains only a vague representation of the true present ridge structure (6.4(b)). An image enhancement can achieve a closer approximation (6.4(c)) of the true ridge structure from a sample.

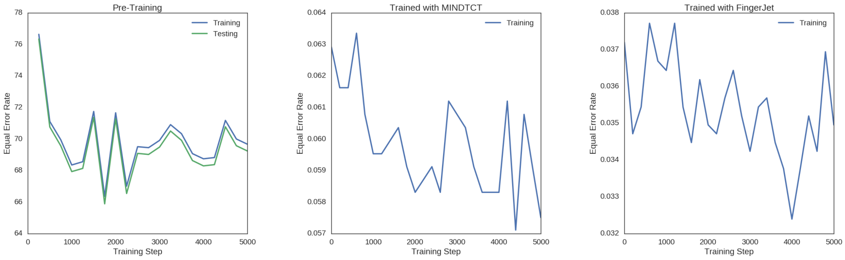
defined setting *Low Quality* for generation of the fingerprint samples. We generated 500 different fingers with 4 impressions each. The entire set was split into two parts: 1,000 fingerprint samples for training and 1,000 fingerprint samples for testing. Figure 6.4 shows an example of a generated fingerprint and the provided ground truth ridge structure.

### 6.4.2 Auto-Encoder Pre-Training

The model was trained first as an Auto-Encoder. Training was stopped after 5,000 training steps. Neither cost on training data nor cost on validation data improved significantly anymore. We therefore assumed, that the model then had learned a reasonable representation for fingerprints. The left image in Figure 6.5 visualizes the development of the cost function for training and validation data over the training steps.

### 6.4.3 Minutia-based Fine-Tuning

The pre-trained model was fine-tuned for 5,000 more training steps. We stopped training then, because the potential of our approach to improve the accuracy and the reliability of the feature extraction was clearly revealed already (see Figures 6.5(b) and 6.5(c)). Figure 6.6 visualizes the impact of our approach on an example on the biometric feature extraction. It shows the fingerprint minutia positions from three different sources. Ground truth minutia positions are marked as red squares. Minutia positions extracted from original samples are marked as green stars. Positions after image enhancement are marked as blue points. The FIE increased



**(a)** **(b)** **(c)**

**Figure 6.5:** The Auto-Encoder was trained first. Pre-training the AutoEncoder was stopped after 5,000 training steps, because the costs function did not decrease anymore (6.5(a)). Afterwards all layers were frozen and excluded from fine tuning, which were responsible for the local and the regional representations. Afterwards, those layers were finetuned, which were responsible for the reconstruction. Equal error rates on the training data tend to improve over time for extractors MINDTCT (6.5(b)) and FingerJet (6.5(c)).

the accuracy of the feature extraction, where the blue points were extracted closer than the green stars to the red squares. Green stars and blue points without a close red square indicate spurious features. Red squares without green stars and blue points indicate features which were missed by feature extraction. Those are the two aspects of the reliability of the feature extraction.

Finally, we wanted to assess the impact of the enhancement on the biometric performance. Figure 6.7 visualizes the Detection Error Tradeoff curves [DET]. DETs have been measured for the training and validation data. The DETs show, that the performance was improved compared to the performance on the original unaltered samples. The performance could be improved by using the fine tuned models for image enhancement for both feature extractors.

The fine tuning improved the biometric performance compared to the pre-trained enhancement on the training data. There was a slight difference between the performances when training was done with another feature extractor than the one which used used for final biometric comparison. Thus, different feature extractors seemed to be better supported by enhancements, which were particularly tailored for them. Enhancing the with a fine-tuned model worked also on test data even though the effect is smaller.

## 6.5 Conclusion

A reasonable method of FIE should improve the suitability of the samples for biometric recognition. This work contributed a dedicated cost function for this task. This cost function takes into account the relevant aspects of biometric feature extraction: The extracted features should be as precise as possible in terms of location and they should be reliable, i.e. no feature should be left out or spuriously introduced. We demonstrated the effectiveness of such a cost function by evaluations on a synthetic data set. This allowed to show up the relation between improved feature extraction and improved biometric performance.

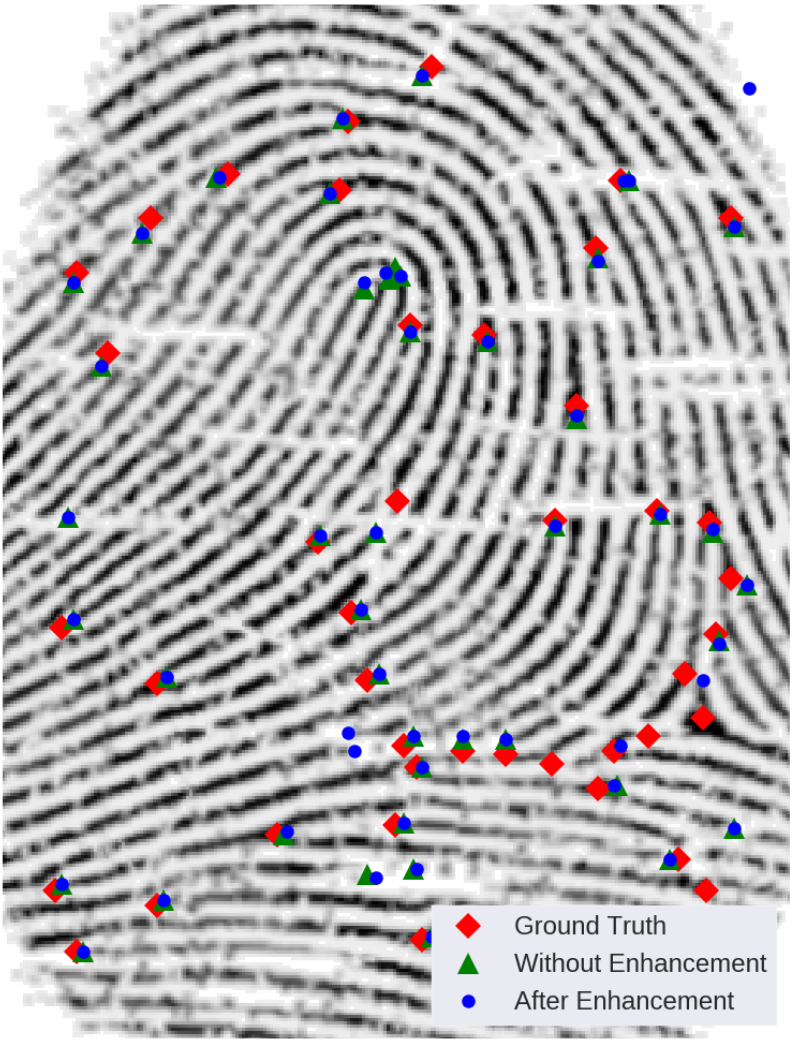


**Figure 6.6:** The ground truth positions of the minutiae are marked as red squares. The positions of minutia after an enhancement (blue dots) should be closer to the ground truth than those extracted from the original images (green stars). Red squares without a close green star or blue dot indicate minutia which are not found by the feature extraction. Green stars or blue dots without a close red square indicate a spurious feature.

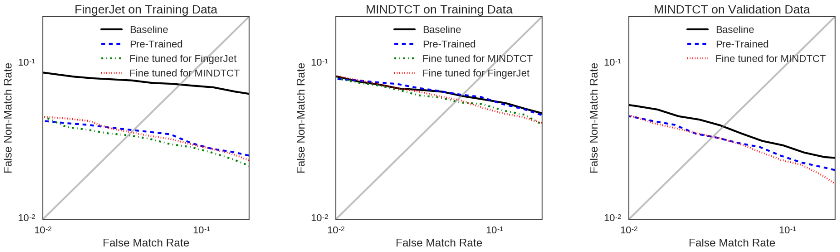


**Figure 6.7:** Detection Error Tradeoff cureves (DET) allow comparison of the biometric performances. Enhancing with a fine-tuned model could improve the performance compared to an enhancement with a model which was only pre-trained. Training for a specific feature extractor was more effective than training on another feature extractor than the one that was used during testing.

# Chapter 7

# Conclusion on Fingerprint Sample Enhancement

Fingerprint samples vary in their biometric quality, i.e. their suitability for an accurate and reliable biometric recognition. To enable the best possible performance of a fingerprint recognition system, fingerprint samples can be enhanced to improve their biometric quality, i.e. their utility for biometric comparisons. It was shown that fingerprint enhancement can improve the biometric performance. The typical measures for biometric quality can even be used for the assessment of the improvement due an enhancement at least to some degree. However, it was also found that not all methods of enhancements can be applied to all kinds of fingerprint data. There is no solution yet, that works best for all kinds of data.

Part II is dedicated to answer the research question $\mathcal{RQ}_1$: *"Can DL outperform classical fingerprint sample enhancements?"* De-Convolutional neural networks can be used to train an enhancement model. Such CNNs can be trained to reconstruct the relevant ridge structure even from strongly disturbed fingerprint samples as they can be found on crime scenes. DL also enables training of models by taking the performance of the biometric feature extraction into account. By doing so, CNNs can even be tailored for the special needs of a given feature extraction process. The opportunity to do so is a paradigm shift in the development of fingerprint sample enhancement. Classical approaches simply were not able to consider this aspect directly. For some types of data, DL based approaches already outperform hand-crafted approaches. In addition, an enormous potential of DL approaches for fingerprint sample enhancement was found. Parallel developments from other researchers also indicate a current superiority of DL based approaches over hand-crafted approaches in the domain of fingerprint sample enhancement.

# Part III

# Orientation Field Estimation

# Chapter 8

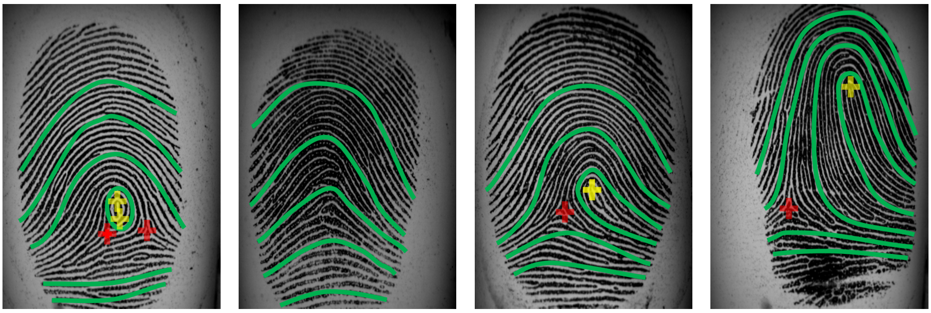
# Introduction

## 8.1 Motivation

A fingerprint orientation field is a representation for the local orientations of a fingerprint's ridge structure (see Figure 8.2). An orientation field represents each position by an orientation $\theta \in [0, \pi)$. The flow of the ridges and therefore the orientation field depends mainly on the positions and relations of so-called *singularities*. There are two types of singularities: *cores* and *deltas*. Cores are those points, where the highest curvature of the ridge structure occurs. Neighbouring ridges are bending around cores. Delta are those points, where three ridge flows seem to merge in a single point. The local orientation cannot be defined neither for deltas nor for cores. Figure 8.1 highlights the positions singularities and the flow of the ridge structure. The count and relation of singularities form typical patterns of orientation fields.

Besides at the positions of singularities, the orientation field can be assumed to be smooth. If a fingerprint was not injured, the local changes in the orientation field are continuous. At the very vicinity of minutiae the local orientation might be slightly disturbed, if one followed the flow of single ridges strictly. However, this aspect is usually ignored in the representation for the sake of the smoothness.

Orientation fields are relevant for many aspects of fingerprint recognition. Besides its position, the *direction* or the *angle* of a minutia is one of its most important features. A minutia's direction can of course be derived from the orientation field at the position of the minutia. The orientation field can also be relevant for fingerprint sample enhancement (see Part II). For example, it is used for the application of reasonable filters during *Contextual Filtering* [110]. The orientation field is



**(a)** Whorl **(b)** Arch **(c)** Right Loop **(d)** Right Loop

**Figure 8.1:** The presence or absence of singularities significantly shapes the orientations fields and builds typical pattern types. Those singularities are *cores* ✚and *deltas* ✚. The green lines emphasize the flow of the ridges around those singularities. The relative positions of the singularities can vary the shape significantly within a pattern type (compare Figures 8.1(c) and 8.1(d)).

also relevant for the process of classical minutia extraction (see Section 1.2 and Figure 8.3). The orientation field is such a remarkable attribute of a fingerprint, that it can even be used as a 1st level feature for biometric comparison. While some fingerprints have similar orientation fields, others can be distinguished by their orientation field at first glance (see Figure 8.1). Therefore orientation fields can be used for exclusion during biometric comparison, i.e. deciding whether two fingerprint samples are non-mated. This can be done for example in *Fingerprint Classification* and *Fingerprint Indexing* (see Chapter 14). Since it is relevant for many applications, an accurate and reliable estimation of the orientation field is a crucial task in fingerprint recognition.

However, the estimation of an orientation field is no trivial task. If the fingerprint ridge structure is not clearly perceivable in some locations, one can only guess the orientation field there. Such guesses usually rely on the orientation field's attribute of smoothness and information from regions, in which orientation in known reliably. Figure 8.5 visualizes typical challenges in fingerprint orientation estimation. Estimation of the orientation field is of course more challenging in regions of higher curvature. Thus, estimations near singularities can be challenging even for high quality samples. If an estimation of the orientation field is not reliable nor accurate, challenges in other processes can arise. For example, erroneous information on minutia angles can be challenging for biometric comparison.

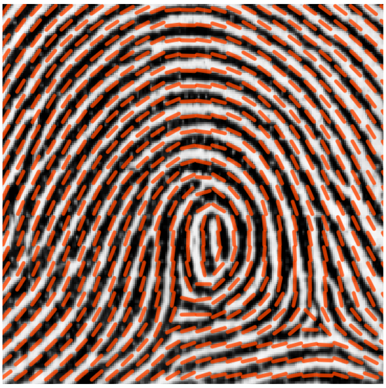


**Figure 8.2:** The orientation field is a representation for the local orientations of the fingerprint ridge structure. Each red line represents the orientation of the local ridges.

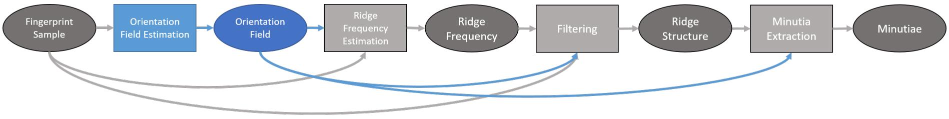


**Figure 8.3:** Orientation field estimation is a crucial task during minutia extraction. The resulting orientation field is used for all following processes of minutia extraction.

## 8.2 State of the Art

### 8.2.1 How to assess algorithms for orientation field estimation?

A lot of ideas were proposed on how to estimate fingerprint orientation fields [26]. But only few proposed methods are evaluated on a common data set and even less with a common metric. Fortunately, there is a relevant benchmark framework available: FVC-ongoing[68]. It provides the possibility to test algorithms for orientation field estimation with common metrics on a common sequestered data set.[1] Thus, FVC-ongoing provides an independent assessment for methods of orienta-

[1] https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/BenchmarkAreas/BenchmarkAreaFOE.aspx

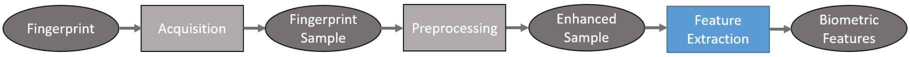
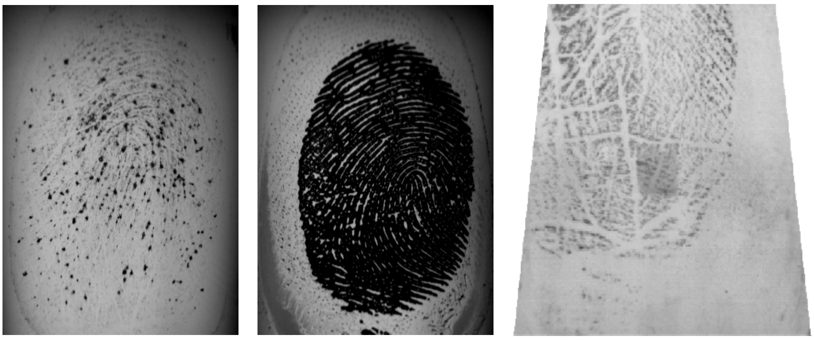


**Figure 8.4:** The orientation field is mainly used in the process of minutia extraction. It can be used to estimate a minutia's directional angle. However, the orientation field can be used in many ways, e.g. it can be used for fingerprint sample enhancement or even for biometric comparison as a 1st level feature.



(a) Dry Skin (b) Moist Skin (c) Scars

**Figure 8.5:** Orientation field estimation is challenging for low quality fingerprints, since the flow of the ridge structure is not clearly perceivable for all locations.

tion field estimation. Over 500 algorithms or rathers versions of algorithms were evaluated at this benchmark so far.[2] Results are reported for ten algorithms. This allows to identify the current state of the art in this domain.

FVC-ongoing uses a dataset of 60 fingerprint samples for evaluation. Ten fingerprint samples are good quality images. Fifty fingerprint samples are of low quality (see Figure 8.5). The low quality fingerprint samples represent typical challenges. All images are livescanner images acquired at a resolution of 500 dpi.

The benchmark uses the *Root Mean Squared Error* (RMSE) as a metric to assess the deviations between estimations and ground truth orientations. Therefore, the ground truth orientation field is sampled at every eighth pixel and compared to the estimation. This results in about 20,000 sampling points for the good quality samples and about 90,000 sampling points for the bad quality samples respectively. RMSE is calculated per fingerprint sample. The RMSE weights every sampling point in a fingerprint sample equally, since the mean is computed over all sampling points. Selection of an appropriate metric is crucial for the evaluation. But iIt is argue, that *"this choice is well suited in most of the feature extraction tasks where orientations are needed (e.g. minutiae detection)"* even though *"other measures may be more appropriate for specific tasks, such as singularity detection"* [50]. However, the quadratic term makes RMSE still sensitive to outliers, i.e. large local deviations between estimations and ground truth. The root allows independence of the any unit or scaling. The ground truth data for the orientations is provided through mark-ups from human experts [50]. The reliability and accuracy

[2] https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/Statistics.aspx

of marked-up ground truth for orientation fields can of course be a critical issue [261]. RMSE is reported separately for good and bad quality fingerprint samples.

Besides the accuracy, other aspects might be relevant for application. The most important side aspect might be the time it takes to estimate the orientation field. If an approach cannot estimate an orientation field in a reasonable amount of time, the approach is simply not applicable. Aspects of timing and memory consumption usually depend on the application scenario. Those aspects might be of highest relevance for biometric recognition scenarios with limited computational resources. However, both aspects might even be neglectable when processing latent fingerprints for offline identification of criminals.

The benchmark FCV-ongoing reports the accuracy of algorithms when dealing with low quality and high quality fingerprint samples, timing and memory consumption. Table 8.1 provides the reported results at FVC-ongoing. State of the art in orientation field estimation therefore can be derived from this table. Estimation of orientation fields is comparatively simple for samples of good quality. Almost all reported approaches achieve an RMSE about 5° on good quality samples. The RMSE of bad quality samples is significantly worse. This rate shall therefore be regarded as the most important aspect for an assessment.

**Table 8.1:** Reported results at FVC-ongoing. Best algorithm per aspect is marked bold. Error metrics $\text{AvgErr}_{\text{BQ}}$ and $\text{AvgErr}_{\text{GQ}}$ assess the deviation of between estimations and manually marked-up ground truth orientations.

| **Algorithm** | **Participant** | **$\text{AvgErr}_{\text{BQ}}$ [°]** | **$\text{AvgErr}_{\text{GQ}}$ [°]** | **Avg. Time [ms]** | **Max Mem. [kBytes]** | **Ref** |
|---|---|---|---|---|---|---|
| *Adaptive-3* | Turroni et al. | 13.27 | 5.93 | 4,772 | 121,936 | [290] |
| *AntheusOriEx* | Antheus Technology | 17.06 | 5.46 | 205 | 34,176 | n/a |
| *ConvNetOF* | Schuch et al. | 8.53 | 5.80 | 6,096 | 939,212 | [259] |
| *DEX-OF* | Schuch et al. | **7.52** | **4.89** | 4,340 | 758,356 | [260] |
| *FOMFE* | Wang et al. | 21.44 | 6.70 | 1,996 | **10,196** | [301] |
| *Gradient* | Ratha et al. | 21.83 | 5.86 | **74** | 42,872 | [240] |
| *LocalDict* | Yang et al. | 9.66 | 6.08 | 5,987 | 67,544 | [319] |
| *MXR* | Zengbo Xu | 11.36 | 5.59 | 2,937 | 11,140 | n/a |
| *OriNet* | Zhejiang University | 8.44 | 6.94 | 1,483 | 1,019,172 | n/a |
| *ROF* | Cao et al. | 11.20 | 5.24 | 762 | 671,984 | [41] |

## 8.2.2 Hand-crafted approaches

Conventional fingerprint orientation field estimation techniques can roughly be divided into two approaches: local estimations and global estimations. A clear classification into global and local approaches is not possible for all algorithms. Sometimes there is a smooth transition between the categories. For instance, global approaches typically incorporate a local approach for an initial estimation. There is a survey on proposed algorithms provided by Biradar and Sarojadevi [26].

With respect to bad quality sample, the model-based approach named *ROF* by the authors, which is using regularized partial differential equations, achieved the highest accuracy of all hand-crafted approaches at FVC-ongoing [41]. This approach models the orientation field by a non-linear partial differential equation. Divergence and curl of the vector field are then used for a regularization. ROF achieved an average RMSE of about $11.2°$ on bad quality samples.

### 8.2.3 Learning-based approaches

Two branches of approaches apply machine learning for the task of fingerprint orientation field estimation. There are approaches using dictionaries and those applying DL.

FVC-ongoing contains a reported result for an approached called *LocalDict*, which is using local dictionaries [319]. Each item in the dictionary represents a typical patch of orientation. An initial estimation for the orientation field is performed. Each location in the fingerprint is then associated with candidate items from the dictionary. The smaller candidate items are then used to assemble the orientation field for the entire fingerprint. LocalDict achieved an average RMSE of $9.6°$ on bad quality samples at FVC-ongoing. This already outperforms the best hand-crafted approach ROF, which achieved about $11.2°$. This approach was also demonstrated on latent fingerprints.

The three best reported results so far with respect to RMSE achieved on bad quality fingerprint samples all apply DL: *DEX-OF*[260], *OriNet*, and *ConvNet-OF*[259]. This is therefore state of the art. Those approaches used CNNs for the estimation of orientation field directly from the fingerprint samples. The best approach DEX-OF achieved an average RMSE of about $7.5°$ on bad quality samples. This outperforms all reported hand-crafted approached significantly.

## 8.3 Contributions

Two approaches for estimation of fingerprint orientation fields are provided. The first approach is called *ConvNet-OF* (see Chapter 9) [259]. Estimation of the orientation field is an estimation of a continuous value. Therefore, application of a regression is the obvious choice. ConvNet-OF applies a regression for estimation of the orientation field. This approach achieved the best accuracy with respect to bad quality samples by the time. The second approach is called *DEX-OF* (see Chapter 10) [260]. DEX-OF applies a classification on this task. By using a technique called *Deep Expectation*[247], classification can even outperform the regression-based approach ConvNet-OF. DEX-OF achieved the best results reported by FVC-ongoing for good quality sample as well as for bad quality samples. This approach can therefore be considered current state of the art in orientation field estimation.

Mark-ups of orientation fields by human experts were assessed for their reliability and accuracy (see Chapter 11) [261]. This assessment revealed, that accuracy and reproducibility of such mark-ups is limited. It also allows assumptions on reasonable lower bound for accuracies. The evaluation yields, that an accuracy of about $5^\circ$ RMSE seems to be a reasonable lower bound for estimation of orientation field estimation for good quality images.

# Chapter 9

# ConvNet Regression for Fingerprint Orientations

**Summary** Estimating the orientation field of a fingerprint sample is a typical regression task. This publication contributes a Convolutional Neural Network for estimation of the fingerprint orientation field. The network's architecture as well as its training are described. Evaluations at benchmark framework FVC-ongoing prove its high performance in accuracy.

This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was presented at the *Scandinavian Conference on Image Analysis* in Tromsø(Norway) in 2017.

[259] P. Schuch, S.-D. Schulz, and C. Busch. Convnet regression for fingerprint orientations. In *Scandinavian Conference on Image Analysis*, pages 325–336. Springer, 2017

**Abstract** Estimation of orientation fields is a crucial task in fingerprint recognition. Many processing steps depend on their precise estimation and the direction of fingerprint minutiae is a valuable information. But especially for regions of low quality the task is not trivial and engineered approaches on local features may fail. Methods that combine local and global features learned from the data are state of the art and benchmarked with the framework FVC-ongoing. We propose to use Convolutional Neural Networks trained in a regression to estimate the orientation field (ConvNetOF). Regression is more accurate than classification in this case. Our approach achieves an RMSE of 8.53° on the Bad Quality Dataset of the FVC-ongoing benchmark. This is the best result reported so far.

## 9.1 Motivation and Introduction

Fingerprint recognition is one of the most wide spread biometric modalities, when it comes to identification and verification of individuals. Recognition algorithms make use of the distinctive features in the fingerprints. Fingerprint minutiae are features, which are typically used for recognition. Minutiae are characteristic points of the papillary ridges, e.g. an ending and a bifurcation [203]. The spatial distribution and relations of positions and directions of minutiae are unique for every finger which allows to distinguish fingerprints.

The direction of a fingerprint minutia is one of its most valuable information for recognition besides its type and position. It directly depends on the local orientation at its location. The orientation field (OF) of the papillary ridges (see figure 9.3(a)) is itself another important feature in fingerprint recognition [203].

Besides this, the OF is relevant information for image enhancement and many processing steps along the workflow of a biometric feature extraction [203]. Deviations between the estimation and the real OF have to be as small as possible for the whole fingerprint area [26]. Otherwise biometric features may not be extracted correctly or spurious features may be generated.

Because of this, an accurate and reliable estimation of the OF is needed for fingerprint recognition. But an accurate estimation is challenging especially for low-quality fingerprint images.

Techniques and ideas for estimating the OF are vast. They can roughly be divided into local and global techniques [26]. Local techniques are based on the very vicinity of every point, e.g. by calculation of local gradients on grey-values in fingerprint images. Those techniques often are not reliable in areas of low quality [26].

In contrast, global techniques usually take benefit of models for the global OF (see figure 9.1(a)-9.1(c) for typical patterns of OFs). The drawback in constructing OFs is that this tends to overly smooth local irregularities and regions of high curvature. In consequence hypothesized models are insufficiently representing the actual OF. Computational complexity in general is higher for global methods than for local ones [26]. Hybrid versions of both try to compensate the drawbacks. However, especially for images of low quality the estimation of the OF is still challenging. The Fingerprint Verification Contest (FVC-ongoing) is providing a benchmark area for fingerprint orientation extraction [68]. As results of this benchmark show, deviations between estimated and real OF are still significantly higher for low quality fingerprint images than for images of higher quality [50]. Closing this gap is one key factor for a more accurate and more reliable fingerprint recognition.

Recently, methods of machine learning, which combine local and global features and furthermore learn directly from the data seem to become a promising solution for OF estimation [319]. In general, techniques which learn from the data, have shown their superiority over engineered techniques in the last decade for various image processing tasks. Techniques from the domain of Deep Learning (DL) and especially Convolutional Neural Nets (CNN) are state of the art at numerous benchmarks, e.g. ILSVR [249]. Significant improvements have been achieved by DL in the domains of Speech Recognition, Signal Processing, Object Recognition, Natural Language Processing, and especially in Multi-Task and Transfer Learning [22].

The versatility of CNNs and Deep Learning techniques enables them to estimate the OF of fingerprints. Our approach is to train a CNN as a regression. This allows to learn an estimation for the continuous valued OF directly from the data.

The rest of the paper is organized as follows: Related work in terms of OF estimation and benchmarking of proposed approaches is discussed in section 9.2. Our suggested approach will be explained in section 9.3. Section 9.4 summarizes the results and conclusions are made in section 9.5. Section 9.6 adds remarks on the findings of this work and gives an outlook on future work.

## 9.2 Related Work

### 9.2.1 State of the Art: Benchmarking

Benchmarks are inevitable for a quantitative evaluation and comparison of approaches. The University of Bologna provides such a public benchmark framework for specific tasks in biometric recognition: *FVC-ongoing* [68]. It also contains a benchmark for *Fingerprint Orientation Extraction* (FOE). The benchmark

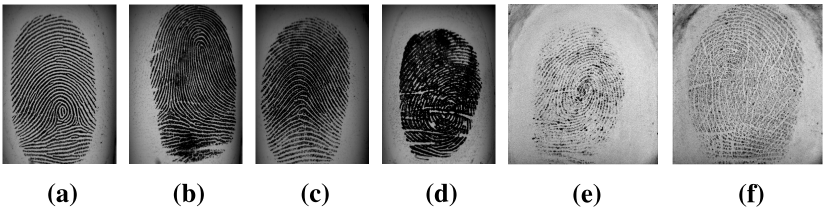


**Figure 9.1:** Figures 9.1(a)-9.1(c) show images of good quality. The orientations differ between fingers and form typical patterns. Figures 9.1(d)-9.1(f) show examples of images with lower quality representing typical challenges. Quality of a fingerprint image can be affected by the moisture of the finger and many other factors. 9.1(d) shows a sample with very moist skin, where the fingerprint in 9.1(e) is rather dry. In addition, 9.1(f) shows scars.

is on-going and it allows to measure performance of algorithms for fingerprint orientation estimation. Implemented algorithms can be uploaded and tested. FVC-ongoing is the only benchmark offering independent measurements on common sequestered dataset and defined metrics for this task. Therefore we report our results based on the quantitative measurements provided by FVC-ongoing.

### Data Set

The benchmark consists of two data sets. Dataset *FOE-TEST* is available for evaluation purposes by the contestants. Dataset *FOE-STD-1.0* is available only to the organizers of the benchmark. Both training and test set are divided into two categories: images of good and images of bad quality. According to their image's quality, the sets are called *Good Quality Dataset* and *Bad Quality Dataset* [48]. For the *Good Quality Dataset* 10 samples are provided, while for the *Bad Quality Dataset* 50 samples are provided (see Figure 9.1 for examples). About 90,000 training data points are provided which represent the local orientation at a single pixel of an eight-fold sub-sampling grid (see Table 9.1).

The images are captured with fingerprint livescanners at a resolution of 500dpi. The *Bad Quality Dataset* shows typical challenges in processing fingerprint images. This set consists of images from fingers with different levels of skin moisture (compare the wet finger in figure 9.1(d) to the dry one in figure 9.1(e)) and the presence of scars in the fingerprint (see figure 9.1(f)). The data is close to what operational data of low quality may look like. The ground truth label data has been produced by manual labelling [50]. The orientation is sampled at an equidistant grid and angles are provided in 256 steps. Labelling is carried out with support

| Set | Name | Number of Samples | Number of Data Points |
|---|---|---|---|
| FOE-TEST | Good Quality Dataset | 10 | 18,946 |
| | Bad Quality Dataset | 50 | 75,812 |
| FOE-STD | Good Quality Dataset | 10 | 19,260 |
| | Bad Quality Dataset | 50 | 89,562 |

**Table 9.1:** FVC provides datasets consisting of a Good and a Bad Quality dataset each.

of a tool introduced by Maltoni et al [48].[1] Additionally to the fingerprint images and the ground truth orientation, a foreground mask is provided. Only OF samples which are in the foreground area will be evaluated.

### Metrics

The four central aspects measured by the benchmark are the deviations between predicted and actual OF achieved on the *Good Quality Dataset* ($\text{AvgErr}_{\text{GQ}}$) and on the *Bad Quality Dataset* ($\text{AvgErr}_{\text{BQ}}$) of *FOE-STD1.0*, memory consumption, and average processing time for each sample. The measure for the OF deviation is the average Root Mean Squared Error (RMSE) observed at all data points. RMSE averages over all sampling points in the fingerprint area in a single sample image. One may argue that deviation might be more important in highly curved regions than in regions of more or less constant orientation, e.g. regions around OF singularities are highly curved. Accurate estimation in those regions is necessary for localization of singular points. In contrast, the benchmark organizers argue that weighting all points equally is suited well for most of the other feature extraction tasks where orientation is needed [50]. The most important measure is $\text{AvgErr}_{\text{BQ}}$ since this metric quantifies the ability of algorithms to handle challenging images.

## 9.2.2 State of the Art: Algorithms

Many ideas for fingerprint OFs estimation have been proposed. A broad survey of OF estimations with qualitative assessments is given e.g. by Biradar et al [26]. But only seven results have been published for FVC-ongoing so far. The two approaches *LocalDict* and *ROF* are performing best in terms of minimizing the

[1] The workflow for labelling is roughly as follows (see [50] for details): A human expert selects a pixel location, which he wants to label. The tool estimates the local orientation by calculating the gradient. The expert may choose to accept the orientation estimate provided by the tool or do a manual correction. A Delaunay triangulation on all labelled points is performed. Each sampling point will be interpolated based on the supporting points of its surrounding Delaunay triangle.

deviation achieved on the *Bad Quality Dataset* of FVC-ongoing. Therefore, those methods are worth a closer inspection and will be described below.

Yang et al provide the best performing algorithm yet called *LocalDict*[319]. They propose to learn dictionaries of OF patterns. The dictionary contains prototypes for local orientation patterns. In a second step, co-occurrence and spatial distribution of the prototypes is learned. Those aspects represent the global structure of fingerprint OFs. Thus, the proposed algorithm combines local and global information. The algorithm first learns a rough estimate of the OF. The locally best fitting prototype is assigned to each point. Finally, corrections of assigned prototypes are performed to optimize likelihood of spatial co-occurrence of the prototypes.

Cao et al proposed an algorithm they call *ROF*. It extracts first an estimation of the OF by the gradient method applied to a root filtered image [41]. The OF is represented as the gradient vector field. In addition, the positions of singularities are estimated. The idea is to smooth the OF while keeping divergence and coherence of the orientation vector field. Intensity of smoothing is varied according to a specific local quality and the distance to a near-by singularity. Thus, areas of high quality and those close to singularities will be smoothed less.

Using Neural Networks and utilizing learning from the data for fingerprint recognition has been suggested previously. Baldi et al already proposed to use a structure like modern Siamese CNNs (without pooling layers) for fingerprint indexing already in 1993 [14]. Zhu et al used a Multi-Layer Perceptron to estimate a 16 step quantization of the OF in 2006 [335]. Olsen et al. used self organizing networks to estimate fingerprint sample quality[225].

Using techniques from DL especially for OF estimation is a more recent development. Sahasrabudhe et al proposed to use Restricted Boltzman Machines (RBM) to estimate fingerprint OFs [250]. An RBMs is probabilistic model which uses a bi-directional neural network. RBMs therefore are not straight feed-forward. An initial OF is estimated and the estimation is vectorized into x and y components. Each component is fed into a separate single-layer RBM. The trained weights of an RBM contain representations for the data used for training which form a basis. Trained RBMs try to approximate the input by this basis. The corresponding output can be interpreted as the best fit to the learned representations. Thus, the output is like a corrected version of the input, which fits best the learned data. The corrected OFs are used to enhance fingerprint images. Finally, performance is measured in terms of the number of spurious minutia extracted by a biometric feature extraction on the enhanced images and in term of the accuracy a biometric comparison algorithm achieved with such extracted features.

The most relevant work with respect to its methodology is an approach by Cao et al which proposes to use a CNN trained as a classifier for orientation [38]. They propose to train a CNN for a classification task. Target labels for the classes are a selection of 128 characteristic OFs, which have to be selected beforehand. Cao et al propose to use engineered noise to corrupt input images. This in turn shall simulate artefacts one in fingerprint images of bad quality.

## 9.3 Proposed Approach

### 9.3.1 Idea

We propose to train CNNs as a regression to estimate the OF in fingerprint images. During the training for a regression, a CNN *model* $\mathcal{M}$ usually learns to minimize the quadratic error between its propagation $\mathcal{M}(\text{inp})$ and the target value $T(\text{inp})$ for a given input inp:

$$\min_{\mathcal{M}} ||T(\text{inp}) - \mathcal{M}(\text{inp})||^2 \tag{9.1}$$

During testing, for an input $\hat{\text{inp}}$ the model $\mathcal{M}$ will create a prediction $\mathcal{M}(\hat{\text{inp}})$.

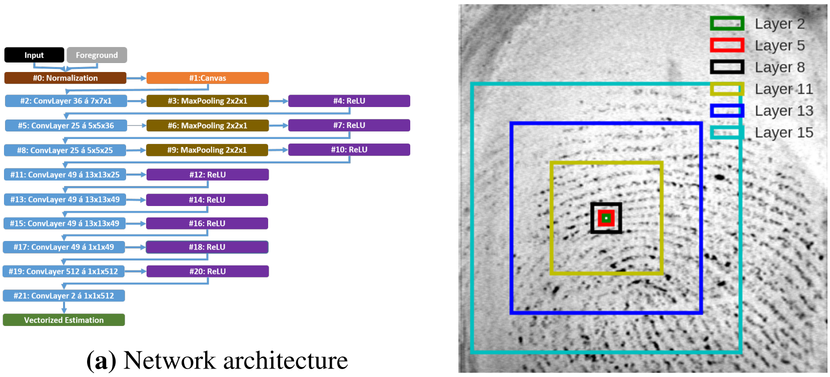
A CNN model $\mathcal{M}$ itself is assembled from multiple sets (layers) of trainable filter kernels. The output of each layer is fed into the next layer of filters. While such models learn simple local features in the first layers, the following layers learn more complex and more global features.

By doing so, our approach utilizes learned local and global characteristics at once. This turned out to be a successful strategy in the *LocalDict* approach. Our approach has three advantages over *LocalDict*. First, our approach does not need an initial estimation of the OF. Second, CNNs use sparse representations for information. This enables a more flexible representation than the one-hot representation used by *LocalDict*. Third, our approach is an end-to-end solution, i.e. input is the raw grey-value image and the corresponding foreground mask and the output is an estimation of the OF. No separate processing steps need to be carried out. No special assumptions about the spatial distribution have to be separately modelled by learned data.

We train the model as a regression on a vectorization of the target orientations. Compared to Cao et al.'s classification approach, regression is a more natural approach for the estimation of continuous values. In addition, no selection of target patterns is necessary.

### 9.3.2 Model Architecture

The proposed CNN has been trained in the framework *Caffe* provided by Jia et al [141].



**(a)** Network architecture

**(b)** Receptive fields of ConvLayers

**Figure 9.2:** Block diagram of the layout of the evaluated model $\mathcal{M}$ which calculates a vectorized estimation of the OF for a given input fingerprint sample and a given foreground mask. $\mathcal{M}$ consists of Normalization, canvas, Convolutional (ConvLayer), Pooling (MaxPooling), and Rectified Linear Unit (ReLU) layers. The receptive field, which is processed from the original image, increases with each layer. While the filters work locally in the first ConvLayers, the last ConvLayers work in a more global range. In this way, local and global features can be combined by a ConvNet.

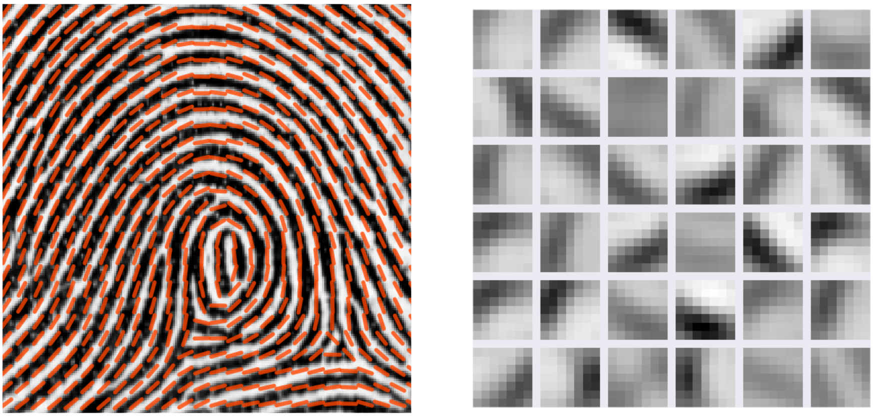
Our approach combines three different types of layers provided in *Caffe*: Convolutional layers, Pooling layers and non-linear transfer layers. Grey values of the fingerprint images are normalized in the foreground area to have zero mean and unit standard deviation while the background is set to zero (Normalization). To enforce the same image dimensions for all training samples the images are embedded into a larger canvas.

Neurons in ConvLayer work like filter kernels (see figure 9.3(b) for trained filters). Pooling layers perform a reduce operation on the local neighbourhoods. They therefore work like sub-sampling. The pooling functions in this approach is the maximum over all local values. The layers are therefore called Max-Pooling. Non-linear transfer units simply apply a non-linear function to each input value, e.g. $ReLU(x) = \max(0, x)$.

Our CNN is designed for the special needs in OF estimation (see figure 9.2(a)). Accurate local estimations are needed as well as regional smoothness and global patterns. It therefore differs from typically very deep cascade of same 3x3 filters. The original fingerprint image is normalized in a Normalization layer. The normalized image is filled into a larger canvas of 576x464 pixels in an Embedding layer. In the following blocks of ConvLayers and ReLU layers are concatenated. Additionally, in the first three blocks MaxPooling layers are used to sub-sample the image dimensions to the provided target dimensions. Filter sizes are designed to cover half of the width of a typical fingerprint ridge. This is done to ensure good local estimations. In the next three blocks larger ConvLayers of dimension 13x13x49 each are used for regional smoothness. All ConvLayers have a striding of 1 and do padding to equalize height and width of input and output. The subsequent combination of MaxPooling and larger kernel leads to a larger receptive field for each output of layer 15, i.e. all pixels within the turquoise area contribute to the output value of layer 15 in figure 9.2(b). This allows to combine the local features to more global ones. Usually, so-called fully-connected layers are used at the end of the cascade of layers. In a fully-connected layer each neuron is connected with every input as in classical Multi-Layer Perceptrons. We use ConvLayers with kernel height and width of 1 as a proxy for such fully-connected layers at the end of our layer cascade [184]. The final layer has two output channels which estimating the vectorized target orientation.

### 9.3.3 Training Algorithm

The CNN in our approach has been trained using a Stochastic Gradient Descent [29]. The cost function is formulated as a quadratic regression on the two-component representation of the orientation $\theta$ at input inp:



**(a)** Local Orientations

**(b)** Filters in first layer

**Figure 9.3:** The training data provides fingerprint images and the hand-labelled ground truth orientation, indicated here as red lines. The filter kernels in the first ConvLayer work like edge filters and do a rough estimation of this orientation.

$$\min_{\mathcal{M}} \left\| \begin{pmatrix} \sin(2 \cdot \theta(\text{inp})) \\ \cos(2 \cdot \theta(\text{inp})) \end{pmatrix} - \mathcal{M}(\text{inp}) \right\|^2 \tag{9.2}$$

The images and ground truth orientation data from FOE-TEST are taken as training data. This data provides 94,758 labelled targets for training. Figure 9.3(a) visualizes a typical representative taken from the training data. The model $\mathcal{M}$ has 1,347,967 parameters. A cost function for large weights is added for a so-called *Weight Decay* to enforce generalisation [164]. Since large weights induce high costs, weight decay punishes over-specialisation of weights and therefore prevents over-training.

Parameters for training are the following: Weight decay factor is 10. Starting learning rate is $10^{-5}$ and adapted according to Inverse Decay policy with $\gamma = 10^{-4}$ and a power of 0.75.

Figure 9.3(b) visualizes the filter kernels of the first ConvLayer after training. The kernels of the first ConvLayer work similar to edge filters. The next ConvLayer recombines those features to more complex features. Figure 9.4 visualizes the output of all ConvLayers for a fingerprint sample after training. The outputs of some filters have only very little absolute values. This is an effect of Weight Decay reducing the energy of unnecessary filter kernels.

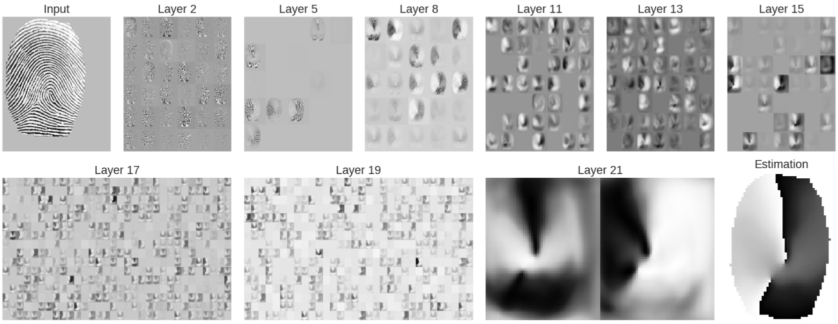


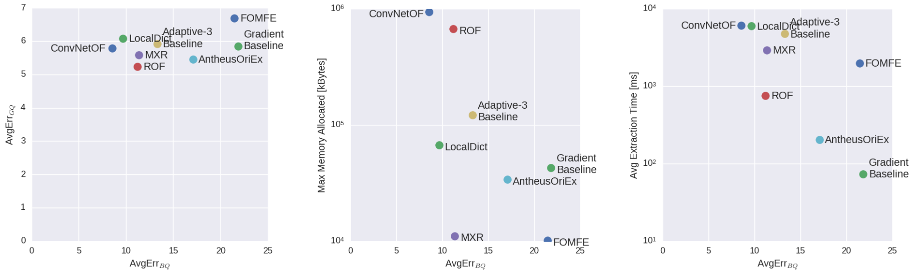
**Figure 9.4:** The output of all ConvLayers for a single input fingerprint sample. Layer 21 represents the estimation for the vectorized OF of the input image. The vectorized OF can be used to calculate the final OF estimation. *Weight decay* can prevent a model $\mathcal{M}$ from over-fitting. As a result of weight decay the output of some kernels has low absolute values.

## 9.4 Results

As mentioned in section 9.2.1, the four central aspects observed in the benchmark are the deviations achieved on the *Good Quality Dataset* and on the *Bad Quality Dataset* of *FOE-STD1.0*, processing time, and memory consumption. Figure 9.5 visualizes the four aspects for all reported results. The benchmark organizers do not provide a overall ranking based on the four aspects.

The reported deviations $\text{AvgErr}_{\text{GQ}}$ for all algorithms do not outperform the baseline algorithm significantly (see table 9.2). Performing well for images of good quality therefore does not seem to be challenging even for simple algorithms. The error rates on this set range from $5.24°$ to $6.7°$ while the baseline algorithm achieves an error rate of $5.86°$. ConvNetOF achieves $5.80°$. The local information extracted by the baseline algorithm is just sufficient.[2] However, the deviation $\text{AvgErr}_{\text{GQ}}$ can be taken into account as a lower bound for the deviations $\text{AvgErr}_{\text{BQ}}$.

[2] Outperforming the baseline algorithm for $\text{AvgErr}_{\text{GQ}}$ might be challenging for a good reason: The *Gradient* algorithm is more or less used to generate the ground truth. One can assume that for good quality images the human editor might consider the initial estimation to be right even though it might show a systematic bias by the algorithm. In contrast, for bad images manual correction might be obvious to the human editor. However, FVC-ongoing still remains the best mean to compare OF estimators.



**(a)** **(b)** **(c)**

**Figure 9.5:** Visualization of the reported results for the benchmark FVC-ongoing FOE. Figure 9.5(a) shows a scatter plot for the deviations achieved on both datasets. Four algorithms outperform the baseline with respect to the deviation of the Bad Quality Dataset. Figure 9.5(b) reveals that memory consumption varies significantly between these top four. A trade-off between Speed and Accuracy can be observed in figure 9.5(c).

To our mind, the most important aspect is the deviation achieved on the *Bad Quality Dataset* for *FOE-STD1.0*. Here deviations vary more than on the *Good Quality Dataset*: they range from $9.66°$ to $21.83°$. Our approach ConvNetOF achieves $8.53°$. This reduces the deviation to about $88\%$ relative to the former best result. Figure 9.5(a) visualizes the error rates for all reported results (compare to table 9.2). On training data *FOE-TEST* ConvNetOF achieves $5.14°$.

Timing and memory constraints strongly depend on the application. In figure 9.5(b) the memory consumption is plotted against the deviation on the *Bad Quality Dataset*. In general, memory consumption is seldom a limitation and may only be a critical issue for systems with very limited memory, e.g. for SmartCards. The consumption of memory varies by orders of magnitudes and ConvNetOF has the highest requirements for memory among all evaluated algorithms.

Figure 9.5(c) reveals a trade-off between deviation and average processing time: The longer the computation time, the more accurate the result. Like in memory consumption, the reported top results vary in their average processing time in an order of magnitude. However, all algorithms are way faster than the time limit of 60s per sample allowed by the benchmark framework. Our approach takes the longest time to process an image. This is due to the evaluation performed on a CPU at FVC-ongoing. However, CNNs are suited for operation on a GPU, which can increase the speed in orders of magnitude: While processing one image takes

| Algorithm | AvgErr$_{BQ}$ [°] | AvgErr$_{GQ}$ [°] | Avg. Time [ms] | Max Mem. [kBytes] | Ref |
|---|---|---|---|---|---|
| *ConvNetOF (R)* | **8.53** | 5.80 | 6,096 | 939,212 | |
| *ConvNetOF (C)* | **8.91** | 6.17 | 6,257 | 943,888 | |
| *LocalDict* | 9.66 | 6.08 | 5,987 | 67,544 | [319] |
| *ROF* | 11.20 | **5.24** | 762 | 671,984 | [41] |
| *MXR* | 11.36 | 5.59 | 2,937 | 11,140 | n/a |
| *Adaptive-3* | 13.27 | 5.93 | 4,772 | 121,936 | [290] |
| *AntheusOriEx* | 17.06 | 5.46 | 205 | 34,176 | n/a |
| *FOMFE* | 21.44 | 6.70 | 1,996 | **10,196** | [301] |
| *Gradient* | 21.83 | 5.86 | **74** | 42,872 | [240] |

**Table 9.2:** Reported results on FVC-ongoing. The table is ordered by the error rate on the bad quality set ($AvgErr_{BQ}$) and only four results outperform the current baseline performance on this aspect. ConvNetOF is evaluated as a regression (R) and as a classification (C). As a regression it performs best among all evaluated algorithms on this data set. Best algorithm per aspect is marked bold. For result of ConvNetOF see https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/AlgResult.aspx?algId=5604

about 6.1s on the benchmark system, it takes only about 25ms of our GPU[3] which is about 244 times faster and would allow processing 40fps.

For comparison, we also trained a model as a classification with the same layout but the last layer as prediction for the 256 orientation classes. With the same error on the training set, this model achieves 8.91° on the *Bad Quality Dataset* and 6.17° on the *Good Quality Dataset* for *FOE-STD1.0*.

## 9.5 Conclusion

We have proposed to use CNNs trained in a regression to estimate the OF of fingerprints. Our approach has been evaluated on the benchmark framework FVC-ongoing, which is the most relevant benchmark for estimation of OF. ConvNetOF achieves a deviation of 8.53° on the *Bad Quality Dataset*. Our approach therefore outperforms all other algorithms in this aspect. The deviation on bad quality images is lowered to about 88% relative to the second best result. This narrows the performance gap between the estimation of OF on images of good and those of bad quality. The performance of ConvNetOF on the *Good quality Dataset* is competitive to the other evaluated algorithms. The model trained as a regression outperforms the model trained as a classification. We found a generalization gap between training and testing.

[3] An NVIDIA GTX 780 has been used for evaluation

In terms of memory consumption our approach has the highest requirements among all evaluated algorithms. Using a GPU it outperforms all other approaches in terms of speed.

## 9.6 Discussion and Outlook

The trained model is likely to be over-sized for this task. Inspection of the trained CNN reveals that some filter kernels may be obsolete. For application it would be reasonable to reduce the size of the CNN. This would not only make it faster and less memory consuming but it would also prevent over-training. However, runtime optimization is out of scope for this work.

Some remarks on the benchmark FOE of FVC-ongoing seem worth mentioning. The number of images (especially for the *Good Quality Dataset*) is small. In addition, the ground truth for the orientation may be biased since it has been edited by a human expert who manually corrects the output of an OF extraction algorithm. Both facts in combination are bad circumstances for learning from the data. Evaluations on larger datasets seem reasonable.

# Chapter 10

# Deep Expectation for Estimation of Fingerprint Orientation Fields

**Summary** This publication contributes a Convolutional Neural Network for orientation field estimation. A technique called *Deep Expectation* allows to outperform regression networks on regression tasks. The network's architecture as well as its training is described. Evaluations at benchmark framework FVC-ongoing prove its high performance in accuracy.
This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was presented at the *International Joint Conference on Biometrics* in Denver (USA) in 2017.

[260] P. Schuch, S.-D. Schulz, and C. Busch. Deep expectation for estimation of fingerprint orientation fields. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 185–190. IEEE, 2017

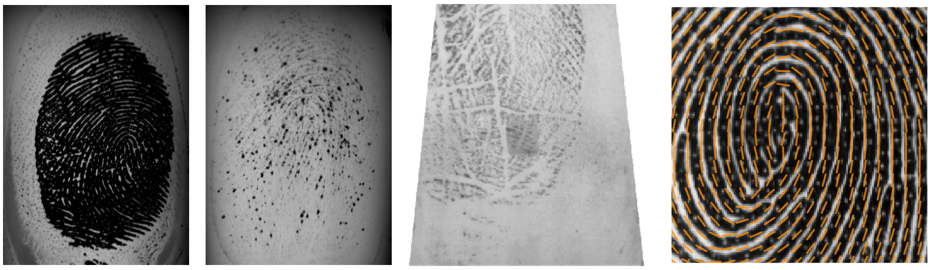
**Abstract** Estimation of the orientation field is one of the key challenges during biometric feature extraction from a fingerprint sample. Many important processing steps rely on an accurate and reliable estimation. This is especially challenging for samples of low quality, for which in turn accurate preprocessing is essential. Regressional Convolutional Neural Networks have shown their superiority for bad quality samples in the independent benchmark framework FVC-ongoing. This work proposes to incorporate Deep Expectation. Options for further improvements are evaluated in this challenging environment of low quality images and small amount of training data. The findings from the results improve the new algorithm called DEX-OF. Incorporating Deep Expectation, improved regularization, and slight model modifications enable DEX-OF to achieve an RMSE of $7.52°$ on the bad quality dataset and $4.89°$ at the good quality dataset at FVC-ongoing. These are the best reported error rates so far.

## 10.1 Introduction and Motivation

*Fingerprint Orientation Estimation* (FOE) is one of the central processes in classical biometric feature extraction for fingerprints [203]. Many processing steps depend on a reliable FOE, e.g. an oriented filtering. Particularly, the direction of fingerprint minutiae (which are the classical biometric features) directly depends on the local *orientation field* (OF). But especially for images of low quality, FOE is no trivial task (see Figure 10.1 for challenging examples). Unfortunately, for those images preprocessing is essential and will only work reasonable well with an accurate and reliable FOE.

Many approaches have been proposed for FOE. But only a few have been evaluated by the FOE benchmark at the independent benchmark FVC-ongoing [68]. The published results show that there is only small variation in precision between the benchmarked algorithms for good quality images (GQ). The precision is significantly lower for bad quality images (BQ) and the precision varies strongly between the different approaches. Deep Learning (DL) has provided promising results in many domains of image processing, their application to FOE is straightforward: The best result for BQ was achieved by the approach *OriNet* which claims to use DL. The second best result was achieved by *ConvNetOF* which uses a convolutional neural network (CNN) trained as a regression [259].

This work assesses five central questions regarding the model *ConvNetOF* by extensive quantitative evaluations. The first question [Q1] is whether the proposed use of Deep Expectation can outperform Regression or Naïve Classification. The



(a) Moist skin (b) Dry skin (c) Scars (d) Orientation Field

**Figure 10.1:** Samples from the bad quality training data (10.1(a)-10.1(c)). Estimation of the local orientation (orange lines in 10.1(d)) on such images can be challenging.

model seems to be oversized when comparing the number of data points for training and the number of trainable parameters (see Tables 10.1 and 10.2). The second question [Q2] is therefore how generalization can be achieved despite this imbalance. The layout of the model differs from approaches typically used today (see Figure 10.3(a)). Thus the next question [Q3] is whether there is a better layout of the network for FOE. The fourth question [Q4] is regarding an appropriate dimensioning for a CNN for FOE. The last questions [Q5] is how the foreground masking provided by FVC-ongoing improves performance.

This work contributes special findings on FOE from the experiments regarding these questions. In addition, it provides findings on various aspects relevant when designing a CNN.

The rest of this paper is organized as follows: Related work is described in Section 10.2. Section 10.3 describes our approach using Deep Expectation. Section 10.4 explains the experiments, which have been conducted to answer the above research questions. The results of the experiments can be found in Section 10.5. The findings will be taken to design and train an improved model called *DEX-OF*. Section 10.6 concludes this work's findings.

## 10.2 Related Work

Neural Networks were used for FOE before. Zhu et al. used a Multi-Layer Perceptron to estimated a 16 step quantization of the OF in 2006 [335]. Sahasrabudhe et al. proposed to use Restricted Boltzman Machines (RBM) for FOE [250]. Even more relevant was an approach by Cao et al., which used a CNN trained as a classifier for FOE [38]. They proposed to train a CNN for a classification task to 128 classes of characteristic OFs.

Only algorithms published at FVC-ongoing can be benchmarked objectively and were therefore of interest in the scope of this work. Most relevant for this work was an approach by Schuch et al. called *ConvNetOF*, which uses CNNs trained as a regression [259]. Yang et al. provided the algorithm in third place at FVC-ongoing called *LocalDict*[319]. Dictionaries of prototypes for local orientation patterns were learned. A rough initial estimate of the OF was locally replaced by the best fitting prototype. Spatial distribution and co-occurrence of the prototypes derived from training data was used to apply some final corrections.

Seven further results have been published for FVC-ongoing so far (see Table 10.3). *OriNet* claims to use DL but lacks detailed information as yet. Cao et al. proposed to smooth the OF while keeping divergence and coherence of the OF and taking into account specific local quality and the distance to near-by singularities [41]. Wang et al. proposed to use 2D Fourier expansion for FOE [301]. Turroni et al. improved the performance of a simple gradient based OF estimation [290] which has been originally proposed by Rathe et al. [240]. No references were published for the two other algorithms *MXR* and *AntheusOriEx*.

## 10.3 Our Approach

The local orientation is a continuous value. Thus, FOE is a typical regression problem. However, classification CNNs can be superior to regression CNNs even on such regression problems [247]. The key is to use a technique called *Deep Expectation*: In order to achieve good regression performance using a classification approach, the regression nature can be encoded by establishing a relation between the classes. The idea is to avoid a winner-takes-all policy by using a weighting strategy (see Figure 10.2).

We propose to use Deep Expectation for FOE. But we need to apply some modifications since we are estimating orientations and we have to deal with the phase shift at the equivalent orientations 0 and $\pi$. The relevant relation between the classes of orientations is of course the angle between them. Therefore, the probabilities for the distinct angle classes are estimated first. The probabilities are then used for a weighted sum over the angles. Finally, this weighted sum can be taken as the output estimation. Using a transformation $T$ for a vectorization of an orientation $\theta$ allows to deal with the phase shift and a straightforward implementation of weighted sums over angles:

$$T(\theta) = \begin{pmatrix} \cos(2 * \theta) \\ \sin(2 * \theta) \end{pmatrix} \tag{10.1}$$

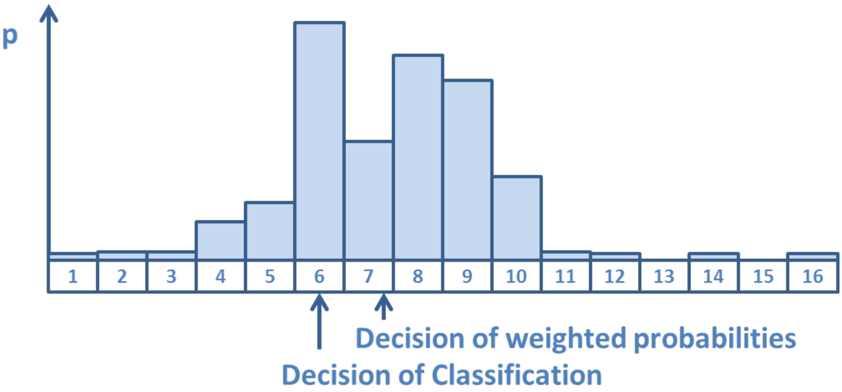


**Figure 10.2:** Example for a 16 class classification problem. When using naïve classication a winner-takes-all policy promotes the class with the highest probality as the most likely class. Deep Expectation calculate a mean weighted bv the estimated probabilities $p$ for the singles classes.

Using the estimated probabilities $p_\theta$ we can build the vectorized weighted sum $W$:

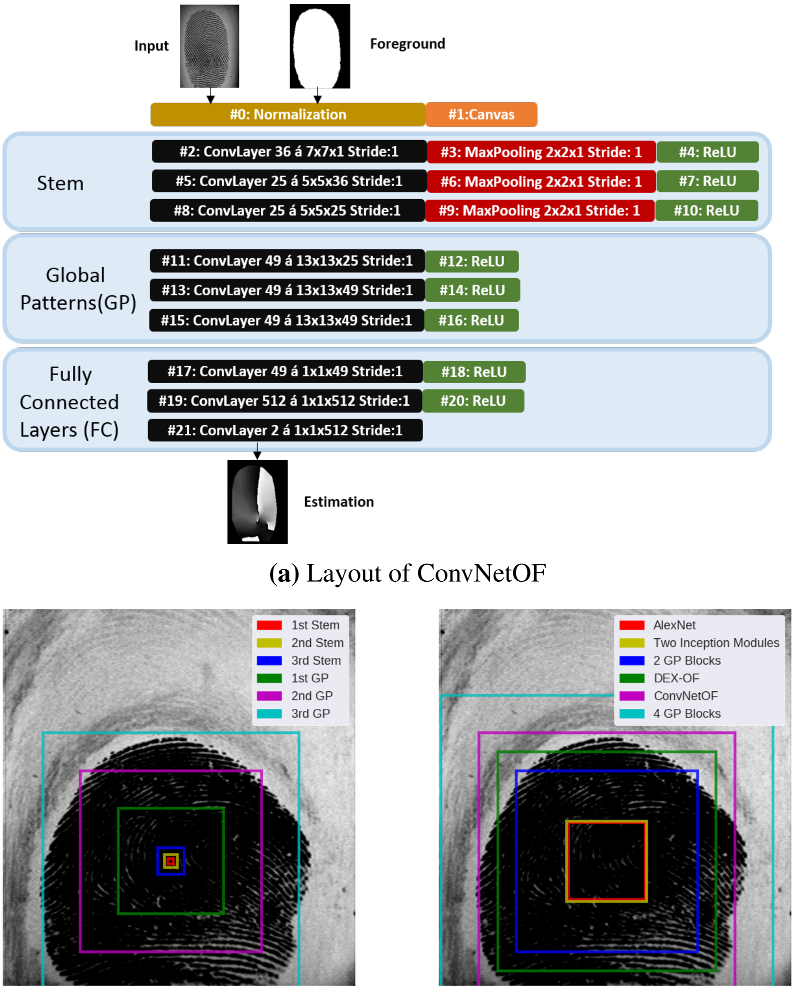
$$W = \sum^{\theta} p_\theta T(\theta) = \begin{pmatrix} w_{\cos} \\ w_{\sin} \end{pmatrix} \tag{10.2}$$

The first component $w_{\cos}$of the weighted sum vector represents the cosine part of the doubled estimated orientation. The second component $w_{\cos}$ represents the sine part respectively. The final orientation estimation $\tilde{\theta}$ can be calculated as follows:

$$\tilde{\theta} = 0.5 \cdot \arctan(w_{\sin}, w_{\cos}) \tag{10.3}$$

## 10.4 Experiments

The CNN ConvNetOF consists of three components (see Figure 10.3(a)). First, a *stem* is built from three blocks. Each block consists of a Convolutional Layer (ConvLayer) with smaller filter sizes, Rectified Linear Unit (ReLU), and a Max-Pooling layer. The stem subsamples the original size to the target evaluation size and shall take care of a good local estimation of the OF and local smoothing. Then, three blocks of large 13x13 ConvLayers and ReLUs reconstruct the *global patterns* (GP layers) from the local estimates. Finally, blocks of 1x1 ConvLayers and ReLUs work like proxies for *fully-connected layers* (FC layers). These take care of the final estimation. The input images were normalized by whitening in the foreground area. All weights were initialized according to Glorot et Bengion [91]. We used the DL framework `caffe`[141] for our experiments.


Input
Foreground
#0: Normalization
#1:Canvas
Stem
#2: ConvLayer 36 á 7x7x1 Stride:1
#3: MaxPooling 2x2x1 Stride: 1
#4: ReLU
#5: ConvLayer 25 á 5x5x36 Stride:1
#6: MaxPooling 2x2x1 Stride: 1
#7: ReLU
#8: ConvLayer 25 á 5x5x25 Stride:1
#9: MaxPooling 2x2x1 Stride: 1
#10: ReLU
Global Patterns(GP)
#11: ConvLayer 49 á 13x13x25 Stride:1
#12: ReLU
#13: ConvLayer 49 á 13x13x49 Stride:1
#14: ReLU
#15: ConvLayer 49 á 13x13x49 Stride:1
#16: ReLU
Fully Connected Layers (FC)
#17: ConvLayer 49 á 1x1x49 Stride:1
#18: ReLU
#19: ConvLayer 512 á 1x1x512 Stride:1
#20: ReLU
#21: ConvLayer 2 á 1x1x512 Stride:1
Estimation
1st Stem
2nd Stem
3rd Stem
1st GP
2nd GP
3rd GP
AlexNet
Two Inception Modules
2 GP Blocks
DEX-OF
ConvNetOF
4 GP Blocks

**(a)** Layout of ConvNetOF

**(b)** Receptive Fields of ConvnetOF **(c)** Receptive Fields of various layouts

**Figure 10.3:** The original layout of ConvNetOF (10.3(a)) included a *stem* for an accurate local estimation and local smoothness. Large ConvLayers on top of the stem learned global patterns and allowed to process information for a large receptive field (10.3(b)). Varying the layout of a model changed the size of its receptive field (10.3(c)). Finally, ConvLayers were used as proxies for fully-connected layers.

| Set | Name | Number of Samples | Number of Data Points |
|---|---|---|---|
| FOE-TEST | Good Quality Dataset | 10 | 18946 |
| | Bad Quality Dataset | 50 | 75812 |
| FOE-STD | Good Quality Dataset | 10 | 19260 |
| | Bad Quality Dataset | 50 | 89562 |

**Table 10.1:** FVC provides datasets consisting of a Good and a Bad Quality dataset each. Each data point is a local orientation on an 8-fold subsampled grid. While FOE-TEST is available for training, FOE-STD is sequestered.

Stochastic Gradient Descent was used for training and learning rate was adjusted by the *Inverse* policy ($\gamma = 10^{-4}$ and power= $0.75$). The cost function to be optimized is the deviation between actual and target orientation at all sampling points. Initial weight decay coefficient was $5*10^{-6}$ and momentum was set to $0.5$. In each training step an entire image was processed. The training data consisted of the fingerprint images and locally sampled OF provided in dataset FOE-TEST (see Table 10.1). To deal with the small number of available training data we used a 10-fold cross validation. By doing so, we could estimate the generalization gap between training and test sets. If not explicitly noted in the experiment description, all training parameter but the learning rate were the same for all experiments. Only the learning rate was adjusted to achieve a similar training error on GQ over training steps. This allowed better comparison of the models. Finally, trained models after different training steps were evaluated at FVC-ongoing.

FVC-ongoing measures four aspects: average *Root Mean Squared Error* (RMSE) for the BQ dataset called $AvgErr_{BQ}$ and $AvgErr_{GQ}$ for the GQ data, average processing time for a fingerprint sample, and the peak memory consumption. The benchmark has three central limitations: algorithms need to be a `win32` application, the size of the executable must not exceed 20 MB, and deviation on the GQ dataset must be lower than 7° RMSE. Thus, FVC-ongoing results are not available for all submissions.

In the following we will explain the series of experiments which will answer the research questions.

### 10.4.1 Q1: Prediction Mode

In this first series of experiments we evaluated, whether a regression or a classification is better suited for FOE. We therefore evaluated four models which differed only in the formulation of the problem of FOE, i.e. how they predict the OF:

- *Regression*: Original layout of ConvNetOF.

- *Naïve Classification*: Layout like ConvNetOF but the final output layer has 256 neuron - one for each orientation class. The training costs were based on cross-entropy of the 256 output neurons.
- *Implicit Deep Expectation (ImplicitDE)*: We took the trained Naïve Classification model and estimated the fingerprint orientation field by Deep Expectation when testing.
- *Explicit Deep Expectation (ExplicitDE*: Same layout as NaïveClassification but the training cost function depended directly on the orientation estimation carried out in a Deep Expectation manner.

### 10.4.2 Q2: Regularization

Large CNNs are prone to overfit on small training data like FOE-TEST. Using additional auxiliary training data would be an obvious choice. Unfortunately, according to our experiments this led to worse performance. The reason for this effect was the difficulty of human experts to markup the ground truth data, which resulted in discrepancies in repetitive assessments by the same expert as well as discrepancies of the markups between experts.

However, regularization seemed reasonable. Zhang et al. gave an interesting view on the regularization in CNNs [328]: Some techniques were designed to explicitly regularize a model while some do it implicitly. We therefore evaluated some of the most common methods of regularization separately:

- *Weight Decay*: Large weights in the neurons may indicate overfitting. The weight decay factor was increased to $5 * 10^{-3}$ as a countermeasure to this.
- *Drop-Out*: Drop-Out is a mean for convergence acceleration [272]. It can also implicitly improve generalization.
- *Affine Data-Augmentation*: Training data was augmented by flipping and/or rotation.[1]
- *Batch Normalization*: Batch Normalization usually accelerates convergence and may also improve generalization.
- *Batch Size*: Increasing the batch size may reduce the risk of overfitting to a single image in each training step.

[1] In contrast to most other cases where data augmentation is used, these affine transformations changed the target labels.

- *Gaussian Noise*: Additive Gaussian Noise $\mathcal{N}_{0,5.0}$ is added to the original set of training images.

- *Gamma*: Inspired by Howard, Gamma Correction $G(I) = I^{\gamma}$ was applied during training, where $\gamma \in [0.2, 5]$ was a random value for each training step [112].

### 10.4.3 Q3: Model Layout

The layout proposed by Schuch et al. for ConvNetOF was not conform with common practices in DL. Today, usually very deep cascades of 3x3 ConvLayers are used, e.g. in ResNet [107]. A common straight forward approach is to fine-tune a pre-trained models for FOE.

- *Inception Modules*: Motivated by *GoogLeNet* from Szegedy et al. we used *Inception* modules [278]. We decided to replace all GP layer by such modules.

- *3x3 Cascades*: We replaced all larger kernels in ConvLayers by a cascade of ConvLayer with kernels of size 3x3.

- *Dimension Reduction*: Inspired by Lin et al. we used 1x1 ConvLayers for dimension reduction between the larger ConvLayers [184].

- *AlexNet*: We chose ImageNet 2012 winning AlexNet proposed by Krizhevsky [163] as a pre-trained model. We dropped the last pooling layers to achieves the appropriate output sampling size, adjusted the size of the output layer, and retrained the model.

### 10.4.4 Q4: Model size

Especially the GP layers and the FC layers seemed to be oversized and worth a closer inspection. We created the following experiments:

- *50% of all Neurons*: We reduced the number of neurons to 50% for each and every ConvLayer but the very last.

- *50% Fully Connected*: The number of neurons in the FC layers was reduced to 50%.

- *Slightly Smaller GPs*: The ConvLayers in the GP blocks was reduced to 32 filters of 11x11 each.

- *2 GP Blocks*: To test whether the model is already too deep, we excluded one of the GP blocks.
- *4 GP Blocks*: To test if a deeper net could still improve performance, we added another GP block.

### 10.4.5 Q5: Is foreground necessary?

FVC-ongoing framework provided a foreground mask for each image. ConvNetOF made use of this information during normalization and for removal of background. It was of interest how much the performance depended on this extra information or if this mask was necessary:

- *No Mask*: No foreground mask was used in the normalization layer.

## 10.5 Results

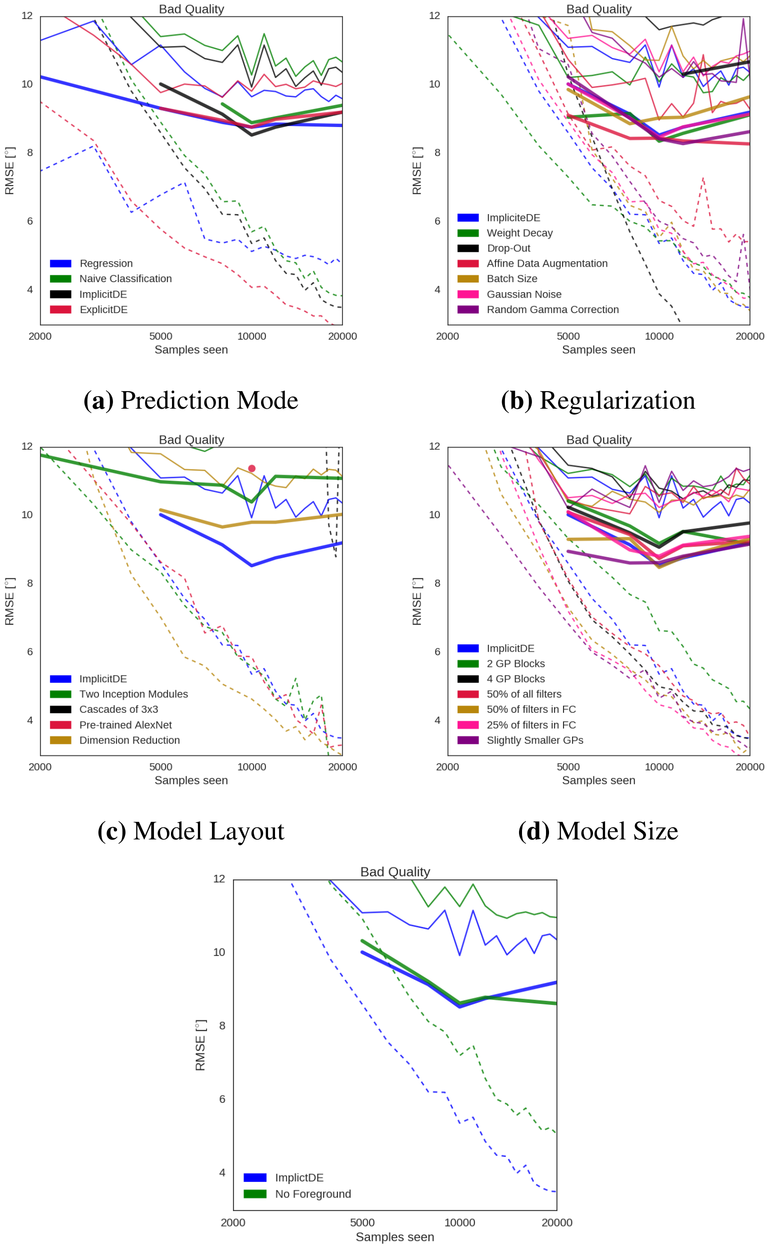
The central results of all experiments can be found in Table 10.2. Figure 10.4 shows the development of RMSE on BQ over training time for all experiments. Using 10 fold cross validation, the mean validation RMSE (solid lines) could be used as an upper bound for RMSE measured at FVC-ongoing (bold lines). As we judge the performance on BQ measured at FVC-ongoing to be the fundamental measure in FOE, results focus on this aspect when discussing the research questions in the following.

### 10.5.1 Q1: Prediction Mode

A model trained as a regression achieved better results on BQ than a model trained as a Naïve Classification (see Figure 10.4(a)). Training the model with explicit usage of Deep Expectation achieved the same performance as the regression model. However, the best result was achieved by the model with implicit usage of Deep Expectation: 8.54°. Therefore, this type of model was used in all of the following experiments.

### 10.5.2 Q2: Regularization

The highest improvement was achieved by Affine Data Augmentation (see Figure 10.4(b)). The BQ RMSE could be reduced to 8.28° at FVC-ongoing. Increasing the Weight Decay parameter also ended up in an improvement. This seemed to be necessary to deal with the imbalance between the number of trainable parameters and the number data points to learn from. Adding Gaussian Noise and applying a gamma correction also slightly improved the performance measured at FVC-ongoing.



**(a)** Prediction Mode

**(b)** Regularization

**(c)** Model Layout

**(d)** Model Size

**(e)** Is Foreground Necessary?

**Figure 10.4:** Comparison of RMSE for evaluated models of the number of samples seen during training. Learning rate has been adjusted to ensure similar training errors for GQ data. Bold lines represent measurements at FVC-ongoing. Due to benchmark requirements some experiments miss this measurement. Dashed lines represent mean training RMSE over the n-fold cross-validations over FOE-TEST and thin lines represent validation RMSE respectively.

**Table 10.2:** Each experiment consists of a ten fold cross validation. For each experiment the mean training and validation RMSE for BQ and GQ. AvgErr$_{\mathrm{BQ}}$ and $AvgErr_{GQ}$ are measured at FVC-ongoing. AvgErr$_{\mathrm{BQ}}$ is the most relevant measure. Experiments on the prediction mode show that ImplicitDE performs best. Therefore experiment in Q2-Q5 and DEX-OF use ImpliciteDE. In addition, DEX-OF makes uses of the findings from Q2-Q5. Improvements are marked bold.

| | **Variation** | **Train BQ** | **Train GQ** | **Valid BQ** | **Valid GQ** | **AvgErr$_{\mathrm{BQ}}$** | **AvgErr$_{\mathrm{GQ}}$** | **#Parameters** |
|---|---|---|---|---|---|---|---|---|
| Q1 | Regression | 5.14 | 3.96 | 9.66 | 6.32 | 8.77 | 5.58 | 1,346,708 |
| | Naïve Classification | 5.71 | 4.28 | 10.29 | 6.66 | 8.90 | 5.88 | 1,476,805 |
| | ImplicitDE | 5.38 | 3.93 | 9.94 | 6.40 | **8.54** | 5.64 | 1,476,805 |
| | ExplicitDE | 4.09 | 3.23 | 9.82 | 6.68 | 8.77 | 5.81 | 1,476,805 |
| Q2 | Weight Decay | 5.45 | 4.10 | 10.09 | 6.34 | **8.36** | 5.65 | 1,476,805 |
| | Drop-Out | 2.91 | 3.01 | 11.77 | 7.97 | 10.32 | 6.91 | 1,476,805 |
| | Affine Data Augmentation | 5.43 | 4.71 | 9.29 | 6.67 | **8.28** | 5.78 | 1,476,805 |
| | Batch Size | 6.49 | 4.82 | 11.15 | 6.78 | 8.87 | 5.74 | 1,476,805 |
| | Gaussian Noise | 5.78 | 4.27 | 10.38 | 6.77 | **8.48** | 5.56 | 1,476,805 |
| | Random Gamma Correction | 5.52 | 4.27 | 10.22 | 6.39 | **8.29** | 5.40 | 1,476,805 |
| Q3 | Two Inception Modules | 5.60 | 3.71 | 12.09 | 6.22 | 10.40 | 4.98 | 869,409 |
| | Pre-trained AlexNet | 5.88 | 3.95 | 13.63 | 5.23 | 11.39 | 5.12 | 11,746,656 |
| | Dimension Reduction | 5.09 | 4.06 | 10.86 | 7.50 | 9.67 | 6.54 | 840,500 |
| Q4 | 2 GP Blocks | 4.35 | 3.03 | 11.19 | 5.71 | 9.17 | 5.35 | 1,070,987 |
| | 4 GP Blocks | 5.01 | 4.20 | 10.80 | 6.96 | 9.08 | 6.09 | 1,882,525 |
| | 50% of all filters | 5.62 | 4.51 | 10.60 | 7.58 | 8.75 | 6.01 | 390,458 |
| | 50% of filters in FC | 4.79 | 3.96 | 10.10 | 6.94 | **8.49** | 5.69 | 1,202,068 |
| | 25% of filters in FC | 4.48 | 3.70 | 10.24 | 7.19 | 8.82 | 6.03 | 1,113,876 |
| | Slightly Smaller GPs | 5.25 | 4.13 | 10.24 | 6.25 | 8.62 | 5.73 | 794,097 |
| Q5 | No Foreground | 5.07 | 3.28 | 10.97 | 6.86 | 8.63 | 6.17 | 1,476,805 |
| | DEX-OF | 5.15 | 3.83 | 6.35 | 4.28 | **7.52** | 4.89 | 523,761 |

Using Drop-Out actually increased convergence speed but even decreased performance. Increasing the number of samples per training step (batch size) did not improve performance.

### 10.5.3 Q3: Model Layout

Using alternative layouts did not achieve an increase in performance. However, there were some remarkable results to each of the approaches.

*AlexNet* was an order of magnitude larger than ConvNetOF in terms of number of trainable parameters. The trained model itself was even too large to comply with the file size limit at FVC-ongoing. Thus, it was simply non-competitive. Luckily, we were allowed by the hosts of the benchmark to evaluate the model anyway by using workarounds. The RMSE achieved on BQ is 11.39°. This might be an indicator that the AlexNet was prone to overfit even more on the small number of data points for training. Another problem with this approach might be that

**Table 10.3:** Reported results on FVC-ongoing. Best algorithm per aspect is marked bold.

| **Algorithm** | **AvgErr$_{BQ}$** [°] | **AvgErr$_{GQ}$** [°] | **Avg. Time** [ms] | **Max Mem.** [kBytes] | **Ref** |
|---|---|---|---|---|---|
| *Adaptive-3* | 13.27 | 5.93 | 4,772 | 121,936 | [290] |
| *AntheusOriEx* | 17.06 | 5.46 | 205 | 34,176 | n/a |
| *ConvNetOF* | 8.53 | 5.80 | 6,096 | 939,212 | [259] |
| *DEX-OF* | **7.52** | **4.89** | 4,340 | 758,356 | |
| *FOMFE* | 21.44 | 6.70 | 1,996 | **10,196** | [301] |
| *Gradient* | 21.83 | 5.86 | **74** | 42,872 | [240] |
| *LocalDict* | 9.66 | 6.08 | 5,987 | 67,544 | [319] |
| *MXR* | 11.36 | 5.59 | 2,937 | 11,140 | n/a |
| *OriNet* | 8.44 | 6.94 | 1,483 | 1,019,172 | n/a |
| *ROF* | 11.20 | 5.24 | 762 | 671,984 | [41] |

the receptive field of the model was essentially smaller than for ConvNetOF (see Figure 10.3(c)). Training an AlexNet from scratch did not work out at all in the number of training steps used for all other evaluated models. Likely it would take far more training steps to achieve a equivalent performance since the model was quite large.

Replacing the GP blocks by *Inception Modules* did not achieve good performance either. The first approach using two *Inception Modules* achieved an RMSE of $10.40°$ on BQ. In contrast, the performance on GQ was remarkably good: RMSE is $4.98°$. According to the findings of Szegedy et al. the usage of fully-connected layers at the end might not be necessary when using *Inception Modules* [278]. When leaving out the fully connected layers the RMSE increased slightly to $10.64°$ for BQ and $5.12°$ for GQ respectively. The receptive field might be too small like it was for *AlexNet*. We did some more variations on this experiments: Enlarging the receptive field by adding three *Inception Modules* failed and therefore was not been evaluated at FVC-ongoing and models with only one *Inception Module* performed worse.

Introducing dimension reduction between the blocks concerned with the global pattern, reduced the number of trainable parameters. However, this decreased performance.

When replacing all ConvLayers by cascades of 3x3 ConvLayers, training did not reach competitive performance.

### 10.5.4 Q4: Model Size

Simply reducing the number of neurons in each layer did not improve performance. The remaining model did not seem to have enough capacity. In contrast, reducing the number of neurons in the fully connected layers to 50% actually improved performance. This reduction was a means to improve generalization. Experiments with further reduction to 25% showed a decay in performance. This was another indication for a lack of capacity of the model.

When slightly reducing the number of neurons and size of the filter kernel in the GP blocks, performance did not drop significantly.

Changing the number of GP blocks did not improve the performance. When using three GP blocks, already most of the fingerprint was in the receptive field. Figures 10.3(b) and 10.3(c) visualize, that it is helpful to take into account global pattern when local information is too vague. When using *4 GP Blocks*, the receptive field was even larger but the number of trainable parameters was likely to be too large to generalize well - like it was for *AlexNet*. When reducing to *2 GP Blocks*, the receptive field of the model was likely to be too small (see Figure 10.3(c)).

### 10.5.5 Q5: Is Foreground Necessary?

If the information of the foreground was not used during normalization of the fingerprint data, the performance decreased only slightly. The RMSE on BQ rose from $8.54°$ for *Implicit Deep Expectation* to $8.64°$. The RMSE on GQ rose from $5.64°$ to $5.99°$ respectively.

### 10.5.6 Final Model: DEX-OF

The results of the earlier experiments were used to design and train a final model. *Implicit Deep Expectation* performed best among the evaluated model types. Therefore, the final CNN will be called *DEX-OF*. *Data Augmentation*, *Gaussian Noise*, and *Gamma* improved the performance and were therefore also be used when training DEX-OF. None of the layout variations achieved a higher performance than the original layout. DEX-OF therefore used the original layout. Reducing the number of filters in the fully connected proxies to 50% was applied. A slight reduction of filter size and in number of neurons in the GP layers achieved almost the same performance as the baseline *Implicit Deep Expectation*. The number of parameters was significantly reduced in this case. Thus, we decided to apply these changes to DEX-OF as well. The total number of trainable parameters was 523,761. Compared to ConvNetOF this was a reduction of about 60%. Memory consumption is reduced by about 30% and average processing time was reduced by about 20%.[2]

By introducing the extra amount of training data and shrinking the CNN it took longer to train the model. However, it still reached acceptable performance within hours.

[2] Using a GPU processing a single sample took 35ms on an NVIDIA GTX 780.

DEX-OF achieved an RMSE of $7.52°$ on BQ while a deviation of $4.89°$ on GQ was achieved.

## 10.6 Conclusions

The experiments designed to answer the five research questions gave some interesting findings. The problem formulation was found to be essential for final performance. Deep Expectation performed best among the prediction modes even though Regression performed better than Naïve Classification on this task. Generalization is a critical issue for CNNs especially for FOE. However, the ratio of numbers of training points and trainable parameters could be compensated by augmenting the training data by affine data augmentation and disturbances like additive Gaussian noise or application of random Gamma correction. Changing the layout of the model to other layouts typically used today did not reach performance of ConvNetOF. But it showed to be important to keep the receptive field of the model large enough to be able to deal with very low quality areas in an image. The influence of the size of the single layers had only little influence on the results. However, reasonable reduction of the size of the layers from ConvNetOF increased the performance slightly and simultaneous reduced the numerical effort. Experiments showed that including the information on the foreground was beneficial even though the benefit was limited.

The number of layers used was significantly smaller than the tens and hundreds of layers which are typically used today. In addition, the training took only a few thousands training steps. Training was done not within weeks but within hours. Therefore, training a CNN for FOE seemed to be easier than others problems to which CNNs are applied to.

We designed and trained a CNN called DEX-OF which achieved RMSE of $7.52°$ on BQ at FVC-ongoing. Compared to OriNet this was an improvement of $10.9\%$. This was an improvement by $11.8\%$ compared to ConvNetOF while the number of trainable parameters was reduced by about $60\%$. Compared to LocalDict the RMSE on BQ was an improvement of about $19\%$. Even though our focus was on BQ, DEX-OF achieved an RMSE of $4.89°$ on GQ. Compared to ROF this was an improvement of $6.7\%$. Both were the best results reported so far.

# Chapter 11

# Intrinsic Limitations of Fingerprint Orientation Estimation

**Summary** Assessing the performance of approaches for fingerprint orientation fields requires ground truth data. In general, such data is not available. It has to be produced by manual mark-up carried out by human experts. This works analyses the reliability and accuracy of such mark-ups.
This publication is joint work with Simon-Daniel Schulz and Christoph Busch. It was presented at the *BIOSIG* conference in Darmstadt (Germany) in 2017.

[261] P. Schuch, S.-D. Schulz, and C. Busch. Intrinsic limitations of fingerprint orientation estimation. *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2017

**Abstract** Estimation of orientation field is a crucial issue when processing fingerprint samples. Many subsequent fingerprint processing steps depend on reliable and accurate estimations. Algorithms for such estimations are usually evaluated against ground truth data. As true ground truth is usually not available, human experts need to mark-up ground truth manually. However, the accuracy and the reliability of such mark-ups for orientation fields have not been investigated yet. Mark-ups produced by six humans allowed insights into both aspects. A Root Mean Squared Error of about 7° against true ground truth can be achieved. Reproducibility between two mark-ups of a single dactyloscopic expert is at the same precision. We concluded that the accuracy of human experts is competitive to the best algorithms evaluated at FVC-ongoing.

## 11.1 Introduction and Motivation

The *Orientation Field* (OF) of a fingerprint is a characteristic feature. It represents the local orientation of the papillary ridges on the fingerprint. The OFs form typical patterns (see Figure 11.1). They are decisive for the orientation of the characteristic points of the fingerprint ridges: the minutiae. Minutiae are the most common biometric features when recognizing fingerprints. Further processing steps may use information of the OF, e.g. image enhancement and automated minutiae extraction. Thus, *Fingerprint Orientation Estimation* (FOE) needs to be *accurate* to allow a precise processing. This makes FOE one of the most important subprocesses in biometric feature extraction from fingerprints [203].

But what does it mean to have an *accurate* FOE? An accurate FOE shall not deviate significantly from the so-called *true ground truth* (GT), i.e. the actual OF. Thus one needs to know GT for a quantitative assessment of an FOE. Unfortunately, the true GT is usually unknown as one does not know the exact OF. To circumvent this lack of true GT, human experts may mark-up GT, i.e. estimate the OF manually and record the estimation.

Whenever estimations are made, they should be questioned and analysed for their accuracy. If in addition humans perform an estimation, reproducibility and whether the humans need expertise can be a critical issue. Despite the fact that FOE is a key aspect in biometric feature extraction, neither accuracy nor reproducibility have been assessed in literature yet. This paper addresses both aspects of FOE by humans.

As a special use case we inspect the benchmark framework FVC-ongoing. It provides the one and only relevant benchmark for quantitative assessment of algorithms for FOE. This of course makes use of a human mark-up of the GT [50].

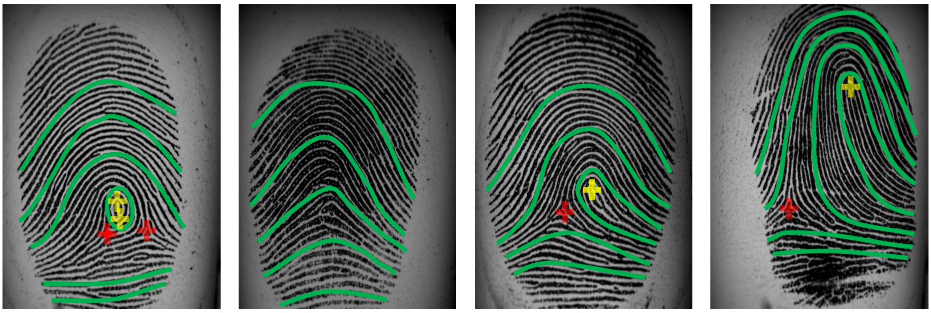
Algorithms under assessment will perform FOE on given fingerprint samples and this estimation is compared to the GT. GT consists of triplets $(x, y, \theta^{\mathrm{GT}})$ representing ground truth orientation $\theta^{\mathrm{GT}}$ at pixel locations $(x, y)$. Let $\theta^{\mathrm{E}}(x, y)$ be the estimated orientation at location $(x, y)$.
Then accuracy can be measured as the *Root Mean Squared Error* (RMSE) over all $N$ sampling points provided in a sample:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_i^{\mathrm{GT}} - \theta^{\mathrm{E}}(x_i, y_i))^2} \qquad (11.1)$$

It is worth mentioning, that the benchmark performs evaluations on two datasets: one data set contains images of good quality (GQ)and the other one contains images of bad quality (BQ). Performance is therefore measured in two scalars: $\mathrm{AvgErr}_{\mathrm{GQ}}$ and $\mathrm{AvgErr}_{\mathrm{BQ}}$ representing the average RMSE over all samples on the single datasets. This splitting takes into account the obvious fact that FOE is a harder task on BQ samples than it is on GQ samples. Published results of FVC-ongoing confirm this assumption (see Figure 11.2). It is surprising to observe that since the FVC-ongoing benchmark was started in 2010, the $\mathrm{AvgErr}_{\mathrm{BQ}}$ has improved significantly over time, while $\mathrm{AvgErr}_{\mathrm{GQ}}$ did not. This may be an indicator for some kind lower bound for RMSE which depends on the benchmark itself. Additionally, this benchmark gives the opportunity to compare the performance of humans against the performance of algorithms tested at the benchmark.
The rest of the paper is organized as follows: Related work is described briefly in Section 11.2. Section 11.3 describes our assessment on the accuracy of FOE. The findings of this paper are summarized in Section 11.5.

## 11.2 Related Work

Some previous work on FOE is relevant for the method proposed in this paper. One of the mark-up tools used in this work was presented by Cappelli et al. [48]. Lodrova et al. have proposed averaging of minutia directions for estimations form multiple experts and define thresholds when consensus on estimations is found [194]. Dactyloscopic examiners were assessed on several aspects: determination of quality [293][226], minutia mark-up [294][295], and identification decisions [291][292]. Oehlmann et al compared algorithms for FOE with two further measures: average deviation (as an alternative to RMSE) and percentage of area with a deviation larger than 15° [224]. This bound of 15° can be considered as a threshold between a reasonable estimation and an unacceptable deviation. Chapman et al. provided a guide for the markup of directions of minutiae [54]. Capelli et al., and Turroni et al. constituted the base for the FOE benchmark at FVC-ongoing [50][290]. The works of Feng et al. and of Gottschlich et al. are



**(a)** Whorl **(b)** Arch **(c)** Right Loop **(d)** Right Loop

**Figure 11.1:** The presence or absence of singularities significantly shapes the orientations fields and builds typical patterns. Those singularities are *cores* (yellow crosses) and *deltas* (red crosses). The green lines emphasize the flow of the ridges around those singularities. The relative positions of the singularities can vary the shape significantly within a pattern type (compare figures 11.1(c) and 11.1(d)).

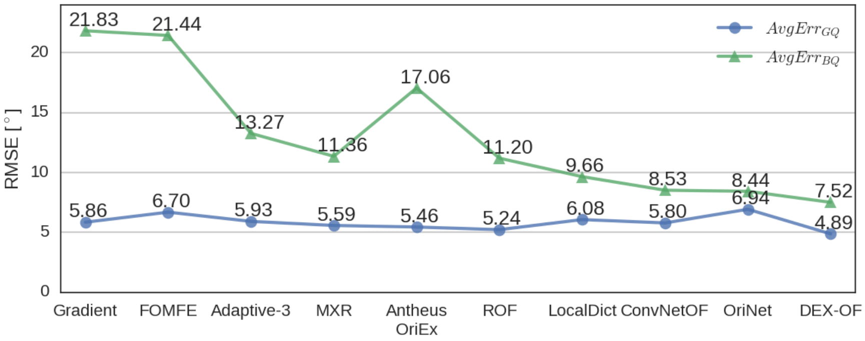
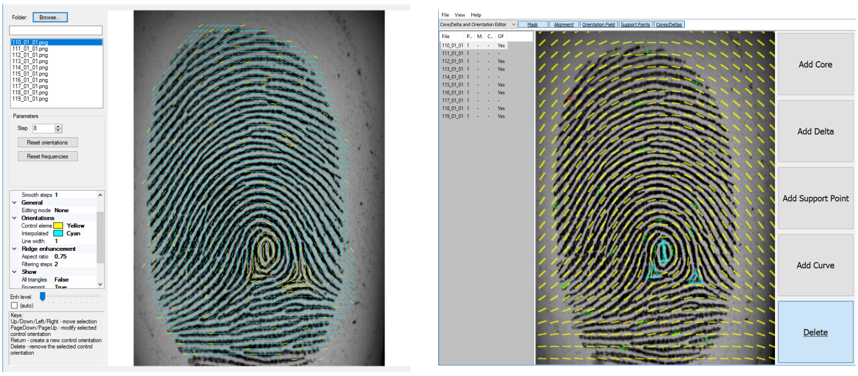


**Figure 11.2:** Algorithms are ordered by their publication date. While $\text{AvgErr}_{\text{BQ}}$ has been improved significantly over time, $\text{AvgErr}_{\text{GQ}}$ stagnates at about 5°.

examples, where manually marked-up OFs were used for assessment of proposed approaches [77][95]. Zhao et Jain used manual markups to separate overlapping fingerprints [331].

## 11.3 Assessment

**Tools** We used two different tools for mark-up. Both differ in the way the mark-up is done and how the OF is constructed from the mark-up.



110_01_01.png
111_01_01.png
112_01_01.png
113_01_01.png
114_01_01.png
115_01_01.png
116_01_01.png
117_01_01.png
118_01_01.png
119_01_01.png

Step 8

| | |
|---|---|
| Smooth steps | 1 |
| **General** | |
| Editing mode | None |
| **Orientations** | |
| Control eleme | 🟨 Yellow |
| Interpolated | 🟦 Cyan |
| Line width | 1 |
| **Ridge enhancement** | |
| Aspect ratio | 0,75 |
| Filtering steps | 2 |
| **Show** | |
| All triangles | False |
| Fingerprint | True |

Enh level
□ (auto)

Keys:
Up/Down/Left/Right - move selection
PageDown/PageUp - modify selected control orientation
Return - create a new control orientation
Delete - remove the selected control orientation

| File | P.. | M. | C.. | OF |
|---|---|---|---|---|
| 110_01_01 | 1 | - | - | Yes |
| 111_01_01 | 1 | - | - | - |
| 112_01_01 | 1 | - | - | Yes |
| 113_01_01 | 1 | - | - | Yes |
| 114_01_01 | 1 | - | - | - |
| 115_01_01 | 1 | - | - | Yes |
| 116_01_01 | 1 | - | - | Yes |
| 117_01_01 | 1 | - | - | - |
| 118_01_01 | 1 | - | - | Yes |
| 119_01_01 | 1 | - | - | Yes |

**(a)** Tool A

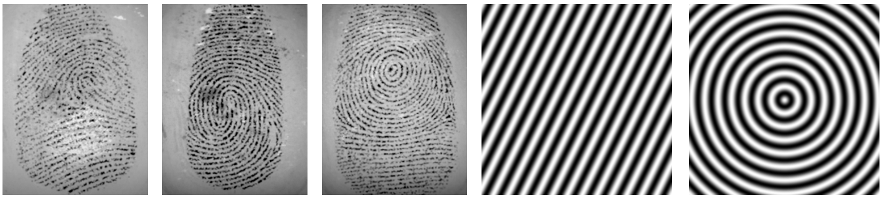
**(b)** Tool B

**Figure 11.3:** Two tools are used for mark-up of OF. Tool A interpolates the OF based on marked support points (yellow lines in 11.3(a)). Additionally to local orientation (green lines), our internal tool B takes into account singularities (cyan triangles and crosses in 11.3(b)) and uses Thin Plate Splines for estimation of the OF.

*Tool A* is called *FingerprintAnalyzer* (see Figure 11.3(a)). It was kindly provided to us by the Università di Bologna. It was the same tool which was used for marking-up the GT at FVC-ongoing. The tool allows a markup at an equidistant grid. It supports the editor by giving an initial estimation for the OF at a selected mark-up point. If the editor does not agree with this estimation, the local OF can be corrected manually. The final OF is calculated as a interpolation based on the marked-up support points. Relevant support points for interpolation are the corners of surrounding triangles of a Delaunay triangulation on the support points. The output is the OF sampled at an equidistant grid of every eighth pixel.

*Tool B* was an internal tool from our team (see Figure 11.3(b)). It allows to mark-up at any point of the sample. In addition to local estimations, this tool allows to mark-up singularities (compare to Figure 11.1). The OF is calculated as a *thin plate spline* (TPS) on the a complex plane based on the singularities. The global shape of the OF is modelled using a Zero-Pole Model. Local deviations from this model can be corrected using control points which use a TPS to interpolate the residual. No initial orientation proposal is provided for the control points, i.e. the orientation of the control points must be set manually. The output is an interpolated OF for every pixel.

**Data Acquisition** Three experts with perennial experience in the domain of fingerprints and three laymen marked-up a total of 15 samples. More reliable results would require more humans involved in the time-consuming mark-up. As we were



**(a)** SFinGe1 **(b)** SFinGe2 **(c)** SFinGe3 **(d)** Lines **(e)** Circles

**Figure 11.4:** The set of samples to be marked-up consists of the ten GQ samples of FOE-TEST, three samples generated with SFinGE (11.4(a)-11.4(c)), and two analytical patterns (11.4(d) and 11.4(e)).

interested in the highest achievable accuracy and best reproducibility, we focused on GQ samples. There were ten GQ fingerprint samples of dataset FOE-TEST provided by FVC-ongoing (file names are "110"-"119"). GT marked-up by a human was available for these ten samples.

In addition, three synthetic fingerprint samples were generated by an external synthesizer tool called SFinGe [46] (see figures 11.4(d) - 11.4(e)). Two pure synthetic samples completed the dataset to be marked-up: straight lines and circle patterns (see figures 11.4(d) and 11.4(e)). Used frequencies were similar to those in fingerprints. For these cases true GT is available.

Both mark-up tools described in Section 11.3 were used for mark-up (see subsection 11.3). Mark-up was repeated in three sessions. At least one day break was made between two consecutive sessions.

In addition, one expert performed two mark-up sessions on the 50 BQ fingerprint samples of FOE-TEST. For these samples manual marked-up GT was available, too.

Dataset FOE-TEST provided GT subsampled at an equidistant grid at every eighth pixel. This sampling rate was the lowest common denominator and was therefore used for all comparisons. In addition, the foreground area containing the fingerprint is provided with the set. As only these areas were relevant, only those were evaluated in the RMSE.

## 11.4 Analysis and Results

**Accuracy** The accuracy of mark-ups for FOE can be assessed most accurately only in comparison to unbiased true GT. We therefore inspected the RMSE achieved on the synthetic SFinGe samples, the lines sample and the circles sample (see Figure 11.4). Table 11.1 revealed that experts performed significantly better than laymen on the task of FOE. They achieved RMSE of 7.8° for all SFinGe samples

**Table 11.1:** RMSE when marking-up with Tool A/B

| Person | Session | SFinGe 1 | SFinGe 2 | SFinGe 3 | Lines | Circles | $\mu_{SFinGe}$ |
|---|---|---|---|---|---|---|---|
| Expert 1 | 1 | 8.9/7.7 | 8.9/8.3 | 6.7/5.6 | 1.6/0.7 | 6.1/2.7 | **8.2/7.2** |
| | 2 | 8.9/7.3 | 8.3/8.2 | 6.5/7.1 | 2.1/0.7 | 6.2/0.9 | **7.9/7.5** |
| | 3 | 7.7/5.3 | 6.9/7.9 | 5.2/5.7 | 1.7/0.7 | 6.7/0.7 | **6.6/6.3** |
| Expert 2 | 1 | 8.5/6.8 | 8.8/7.4 | 8.7/6.1 | 1.7/0.7 | 8.3/1.3 | **8.7/6.8** |
| | 2 | 7.8/6.9 | 7.8/8.2 | 7.2/5.4 | 1.3/0.7 | 6.9/0.6 | **7.6/6.9** |
| | 3 | 8.5/6.1 | 9.0/7.8 | 8.6/5.9 | 3.4/0.7 | 6.6/0.6 | **8.7/6.6** |
| Expert 3 | 1 | 9.4/9.2 | 10.0/7.4 | 6.6/6.3 | 2.4/0.7 | 5.1/0.8 | **8.6/7.6** |
| | 2 | 8.3/9.5 | 8.6/8.5 | 5.6/6.5 | 2.6/0.7 | 2.9/0.7 | **7.5/8.2** |
| | 3 | 8.3/9.6 | 6.6/8.7 | 4.7/5.9 | 1.5/0.7 | 2.2/1.1 | **6.5/8.0** |
| Layman 1 | 1 | 12.3/21.4 | 9.7/13.8 | 7.8/19.8 | 2.2/0.7 | 4.8/6.3 | **9.9/18.3** |
| | 2 | 17.0/23.6 | 9.4/13.2 | 7.9/11.8 | 2.5/0.7 | 8.4/6.8 | **11.5/16.2** |
| | 3 | 11.7/13.0 | 10.8/13.5 | 9.5/8.0 | 1.9/0.7 | 8.1/8.7 | **10.7/11.5** |
| Layman 2 | 1 | 10.7/11.3 | 7.5/12.4 | 7.6/8.3 | 1.5/2.8 | 5.0/7.3 | **8.6/10.7** |
| | 2 | 9.5/11.5 | 8.2/13.4 | 5.2/6.5 | 2.0/3.5 | 5.4/6.5 | **7.6/10.5** |
| | 3 | 10.8/14.7 | 8.2/11.2 | 8.3/8.4 | 2.5/0.0 | 5.4/6.6 | **9.1/11.4** |
| Layman 3 | 1 | 10.1/9.4 | 9.8/10.4 | 5.8/8.7 | 1.9/2.1 | 7.6/5.9 | **8.6/9.5** |
| | 2 | 8.2/8.5 | 9.0/9.5 | 7.8/6.6 | 4.4/0.3 | 5.6/3.9 | **8.3/8.2** |
| | 3 | 10.6/10.8 | 8.5/11.1 | 7.9/9.1 | 2.6/2.8 | 5.6/2.6 | **9.0/10.4** |
| $\mu_{Experts}$ | **all** | **8.5/7.6** | **8.3/8.0** | **6.6/6.1** | **2.0/0.7** | **5.7/1.1** | 7.8/7.2 |
| $\mu_{Laymans}$ | **all** | **11.2/13.8** | **9.0/12.1** | **7.5/9.7** | **2.4/1.5** | **6.2/6.1** | 9.3/11.9 |
| $\mu_{All}$ | **all** | **9.8/10.7** | **8.7/10.0** | **7.1/7.9** | **2.2/1.1** | **5.9/3.6** | 8.5/9.5 |

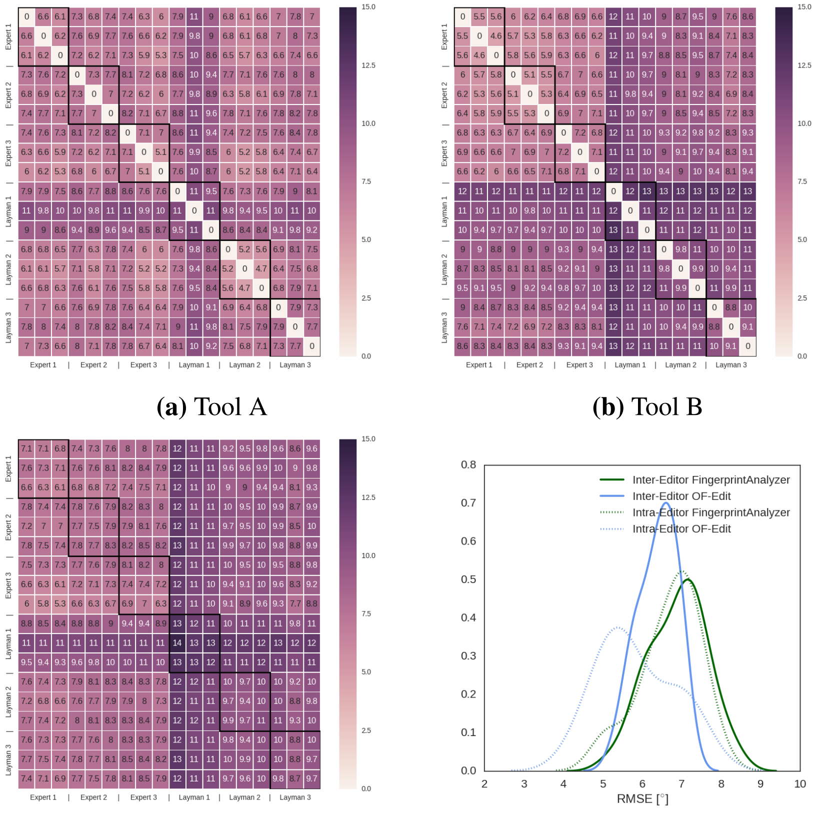
when the tool A was used. When tool B was use, 7.2° was achieved. These performances was better than the RMSE of 9.3° and 11.9° respectively achieved by the laymen. Expertise in the domain of fingerprint recognition was therefore necessary to produce a more reliable mark-up.

The best single mark-up session for all SFinGe samples achieved RMSE of 6.2°. The RMSE achieved for the lines sample showed that this task can be performed with high accuracy. Tool B could be used to better approximate the circles due to the capability to mark-up cores.

**Gaining Expertise** The development of the RMSE over the consecutive sessions gave insight, whether FOE is a task which could be learned fast. Surprisingly, laymen did not improve constantly over time. Despite this, the RMSE for the experts tended to improve over time. We assumed this effect did not reflect an improvement in the task of FOE itself. It reflected the fact that the experts got used to the tools and thus became able to express their knowledge of OF better with the tools.

**Table 11.2:** RMSE against the alleged ground truth provided in dataset FOE-TEST (file names "110"-"119") when marking-up with tool A/tool B. The lowest RMSE achieved over all session is 5.2°.

| Person | Session | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | $\mu_{110-119}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expert 1 | 1 | 7.6/7.3 | 4.9/6.2 | 5.5/6.1 | 7.7/7.0 | 7.3/7.4 | 5.3/7.0 | 5.9/6.3 | 5.5/6.6 | 7.6/6.8 | 5.2/6.3 | **6.2/6.7** |
| | 2 | 6.6/5.7 | 6.7/6.4 | 6.5/5.8 | 7.9/7.0 | 8.3/6.5 | 5.0/6.4 | 7.4/5.4 | 5.9/7.4 | 7.0/7.0 | 5.1/5.2 | **6.7/6.3** |
| | 3 | 5.0/4.9 | 4.5/5.7 | 5.3/6.1 | 7.2/7.0 | 7.8/7.1 | 5.0/6.3 | 5.5/5.1 | 5.1/6.5 | 6.4/7.2 | 4.3/5.7 | **5.6/6.2** |
| Expert 2 | 1 | 6.8/6.7 | 5.8/6.1 | 7.4/9.5 | 9.9/7.3 | 8.7/9.5 | 4.8/7.9 | 6.1/5.5 | 5.5/8.9 | 6.4/7.9 | 5.3/5.3 | **6.6/7.5** |
| | 2 | 6.9/5.3 | 4.9/6.6 | 5.7/9.2 | 8.0/6.8 | 7.6/8.5 | 4.3/6.5 | 5.6/6.1 | 5.4/7.9 | 6.7/8.1 | 5.0/5.1 | **6.0/7.0** |
| | 3 | 6.6/6.5 | 6.6/5.9 | 6.8/10.3 | 8.7/9.2 | 7.9/10.1 | 4.8/6.8 | 6.3/5.4 | 6.1/7.5 | 5.9/7.9 | 7.4/6.2 | **6.7/7.6** |
| Expert 3 | 1 | 6.4/6.5 | 6.4/6.9 | 6.4/7.7 | 8.5/8.5 | 8.1/10.3 | 6.6/6.7 | 6.3/8.4 | 8.6/8.7 | 6.7/8.1 | 6.2/6.1 | **7.0/7.8** |
| | 2 | 5.1/7.0 | 4.6/7.7 | 4.8/6.1 | 6.3/8.9 | 6.6/7.7 | 4.2/5.5 | 5.4/6.7 | 7.1/8.6 | 5.5/8.9 | 4.4/5.7 | **5.4/7.3** |
| | 3 | 4.9/7.4 | 4.5/6.0 | 4.5/6.1 | 5.8/7.7 | 6.7/8.7 | 4.9/6.0 | 5.4/6.6 | 4.4/8.2 | 6.3/7.4 | 4.8/5.7 | **5.2/7.0** |
| $\mu_{\text{Experts}}$ | **all** | **6.2/6.4** | **5.4/6.4** | **5.9/7.5** | **7.8/7.7** | **7.7/8.4** | **5.0/6.6** | **6.0/6.2** | **6.0/7.8** | **6.5/7.7** | **5.3/5.7** | **6.2/7.0** |



**(a)** Tool A

**(b)** Tool B

**(c)** Between Tools

**(d)** KDE of RMSE

**Figure 11.5:** RMSE between all sessions of all experts and laymen. The block diagonal matrix is highlighted by black squares. Those contain the comparison between all sessions of a single person and therefore allow inference on reproducibility of mark-ups.

**Humans vs Algorithms** Table 11.2 contains the RMSE achieved against the GT provided for the samples of FOE-TEST. This allowed to compare the performance of humans against the capabilities of those algorithms evaluated at FVC-ongoing. The mean RMSE $\mu_{110\text{-}119}$ for all experts achieved with the tool A is 6.2° and 7.0° for the tool B respectively. It is worth mentioning, that this was opposite to the higher accuracy against the true GT from the synthetic images when using tool B. This was likely due to the fact, that tool A was used to mark-up the GT. Thus, the results might slightly be biased by the mark-up tool. The best RMSE over all samples $\mu_{110\text{-}119}$ was achieved by expert 3 with the tool A: $5.2°$. This was competitive to the best algorithm at FVC-ongoing (see Figure 11.2).

As lower bounds for BQ samples were of interest, too, we performed some extra assessments. One expert additionally performed two mark-up sessions on the 50 bad quality images of dataset FOE-TEST. The expert achieved a RMSEs of 8.4° in the first and 8.3° in the second session against the alleged GT when using tool B and 11.0° and 9.6° with tool A respectively. The tool B might therefore be more appropriate for mark up of bad quality images. However, this accuracy was competitive to the best algorithm at FVC-ongoing which is called *DEX-OF* [260].

**Local Deviations** The distribution of deviations was not uniform for every sampling point. Figures 11.6(a) and 11.6(b) visualize the degree of dissent on local orientations for all experts on a single sample. Let $\theta_i^{\mathrm{E}}(x, y)$ be the local estimation at location $(x, y)$ from mark-up $i$. Then the local dissent $\delta(x, y)$ can be measured as the mean deviation from an averaged estimation $\mu_\theta(x, y)$ over $M$ mark-ups:

$$\mu_\theta(x, y) = 0.5 * \arctan\left(\frac{\sum_{i=1}^{M} \sin(2 \cdot \theta_i^{\mathrm{E}}(x, y))}{\sum_{i=1}^{M} \cos(2 \cdot \theta_i^{\mathrm{E}}(x, y))}\right) \tag{11.2}$$

$$\delta(x, y) = \frac{1}{M} \sum_{i=1}^{M} \left|\measuredangle(\theta_i^{\mathrm{E}}(x, y), \mu_\theta(x, y))\right| \tag{11.3}$$

The more intense a block was colored red, the larger was the dissent. Not surprisingly, the dissent was larger in the vicinity of singularities than it was in regions of low curvature. The local distribution of dissent was similar for both tools (see Figure 11.6(c)). The area of dissent near singularities was larger for tool B than it was for tool A (yellow circles). Due to the fact that singularities could be marked-up with tool B, slight deviations in position of singularities led to larger areas of dissent. Relevant deviations can also be found where curvature has saddle points, i.e. where the ridges change their bending (blue rectangle). Additionally, there were deviations at those points, where experts had to decide between smoothness of the OF and high fidelity to local changes of the OF (green circle). This was more an individual bias than it was a critical deviation.

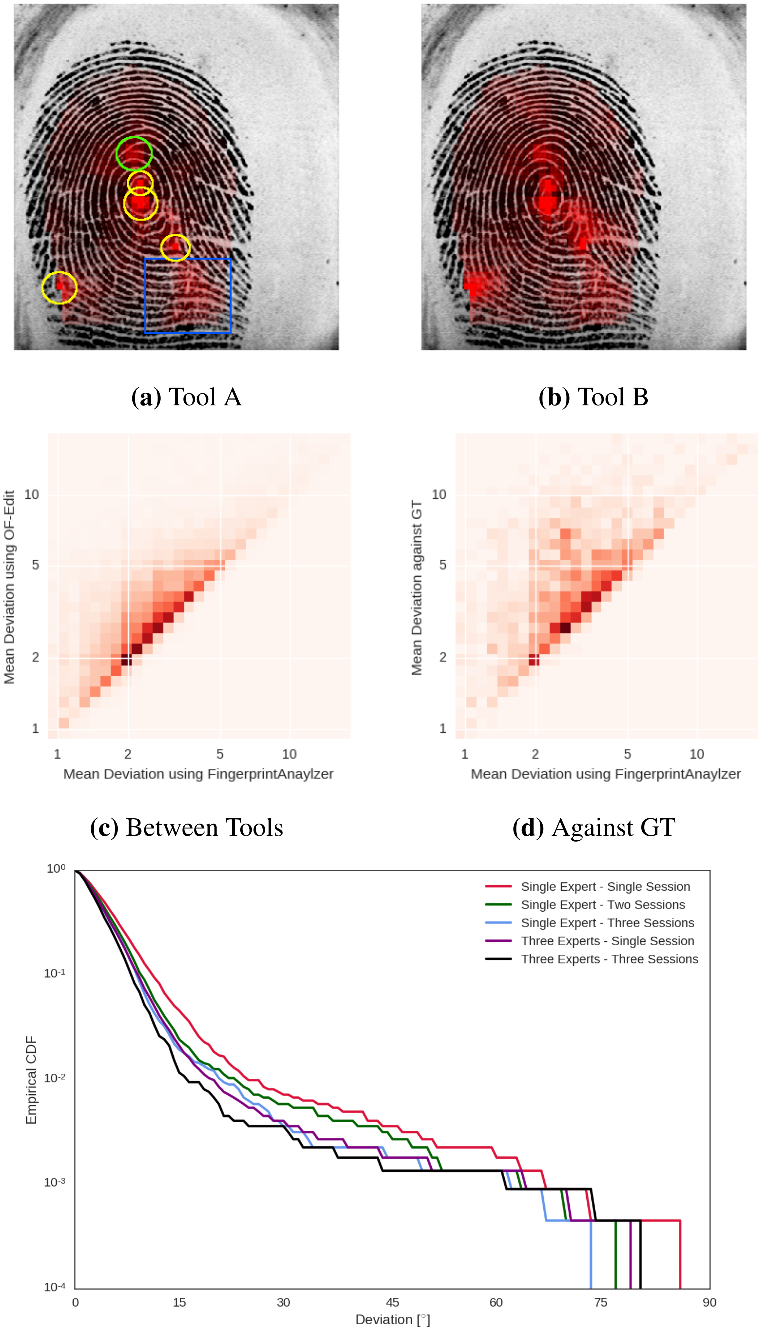
The local deviation among the experts from their estimated mean was strongly correlated to their mean deviation against the GT on the three samples generated with SFinGE. Pearson's correlation coefficient between both mean deviations is 0.8. Therefore, it is likely that dissent among multiple mark-ups will coincide with deviations from true GT.

**Reproducibility** Whenever humans are involved in processes, reproducibility is an important issue. Single mark-up sessions of the human editors were compared against each other to assess this aspect. Figures 11.5(a)-11.5(c) visualizes the RMSE between all mark-ups made by the six human editors. Since also RMSE between all sessions of a single person were included in this graphic, it contains information regarding reproducibility. In general, experts achieved lower RMSE between their sessions than the laymen did. This holds except for layman 2 when marking up with tool A. This good reproducibility needed to be put into perspective of significant higher deviation against true GT (see Table 11.1). However experts could achieve RMSE between 5° and 7° between two mark-ups. Surprisingly, these accuracies were only slightly better than the accuracies between the particular experts. This was an indicator that the single mark-ups were good estimations of the true OF. The RMSE between the two sessions on the BQ samples was 11.7° when using the tool A and 7.6° when using the tool B.

**Approximating True GT** It seemed, that the mark-ups could be interpreted as true GT disturbed by some *noise*. If the noise is mean-free, averaging mark-ups will reduce the influence of noise. Figure 11.6(e) visualized the empirical cumulative density function of deviations between $\mu_\theta$ and the true GT of the SFinGe samples. The more mark-ups involved in averaging, the lower was the deviation against the true GT. There was no significant difference between averaging all three mark-ups of one expert and averaging one session each from all three experts.

## 11.5 Conclusions

By extensive and time consuming mark-up of OFs, we investigated questions regarding FOE when performed by humans. We found that expertise in fingerprints increases the accuracy of marked-up OFs. Experts achieved an RMSE of about 7° compared to true GT. Averaging over more than one mark-up increased the accuracy. Inspection of multiple mark-ups of a single expert showed, that mark-ups could be produced at similar values of RMSE. These values were, therefore, interpreted as rough lower bounds for a reasonable accuracy at FVC-ongoing. When humans were compared to the alleged GT at benchmark FVC-ongoing, they achieved roughly 5° on GQ samples and about 8.4° on BQ samples respectively. This was competitive to the best algorithms evaluated by FVC-ongoing.



**(a)** Tool A

**(b)** Tool B

**(c)** Between Tools

**(d)** Against GT

**(e)** Averaging Mark-ups

**Figure 11.6:** The local dissent among experts on FOE (red tinting in Figures 11.6(a) and 11.6(b)) is similar for both tools. Dissent is strong near singularities (yellow circles), saddle points of curvature (blue rectangle), and where the experts need to choose between local fidelity and smoothness (green circle). Where dissent is large among the expert, the deviation to true GT is large, too (11.6(d)). Averaging over more than one mark-up can reduce such deviations (11.6(e)).

## Acknowledgement

The authors would like to thank the team from Università di Bologna for providing us with their mark-up tool. Additionally, we would like to thank all experts and laymen involved in the mark-up of the OFs.

# Chapter 12

# Conclusion on Orientation Field Estimation

The orientation field is an essential feature of each fingerprint. It is so informative, that it can even be used as a 1$^{\text{st}}$ level feature for biometric recognition. Orientation fields are relevant for many process. This feature can be used for Fingerprint Classification and Fingerprint Indexing. Estimation of the orientation field is a crucial task when processing fingerprint samples. Orientation fields are also important during minutia extraction, since the direction of a minutia is one of its most important features. They can also be relevant for fingerprint sample enhancement. Thus, there are many processes, which rely on an accurate estimation. However, estimation of the orientation field is no trivial task especially when fingerprint samples are of low quality.

Part III is dedicated to answer the research question $\mathcal{RQ}_2$: *"Can DL be used for a better orientation field estimation?"* The benchmark framework FVC-ongoing provides a benchmark for assessing the module of orientation field estimation. It therefore helps answering the research question. DL approaches significantly outperform all hand-crafted and learning-based approaches at this benchmark. The proposed approach DEX-OF achieved an RMSE of $7.5°$ for bad quality fingerprint samples and an RMSE of $4.5°$ for good quality fingerprint samples respectively. This outperforms all hand-crafted approaches significantly and can be considered as the new state of the art. As true ground truth data for orientation fields is unknown, data marked-up by human experts are used for assessment. A lower bound of $5°$ for the accuracy of such mark-ups was found. DL-based approaches therefore already approach the limits of reasonable accuracies in orientation field estimation.

# Part IV

# Efficient Processing Structures

# Chapter 13

# Introduction

## 13.1 Motivation

There are usually two main requirements on fingerprint recognition systems: *biometric performance* and *speed*. If a fingerprint recognition system is not accurate enough for a given scenario, it is simply not applicable. If a fingerprint system does not work fast enough for a given scenario, it is not applicable as well. Thus, these two properties are of highest relevance for a fingerprint recognition system.

Even though fingerprint recognition is a rather straight forward workflow, it is a quite complex process executed in many sub-systems that interact with each other. Each sub-system can be a bottleneck with respect to the biometric performance or speed. Thus, there are many aspects, which can be improved on.

When dealing with large datasets, both speed and biometric performance are very important. Sophisticated minutia-based algorithms for the *thorough fingerprint comparison* of pairs of samples enable highly accurate recognition systems [310]. However, such a thorough comparison of two minutiae point clouds can be a tedious job. Thorough minutiae comparison of probe samples against very large biometric reference databases in a an exhaustive, brute force manner might take too much time. Recognition systems like Aadhaar will deal with more than a billion of individuals [84].

There is no need for an exhaustive search of thorough comparisons against an entire biometric reference database. Some pairs of fingerprint samples can be decided to be non-mated at first glance, e.g. by application of *pre-selection algorithms* like *Fingerprint Classification* or *Fingerprint Indexing*. Fingerprint Classification makes use of *Henry pattern types* (see Figure 13.1 and also Part III). Only pairs of samples sharing the same characteristics are worth a thorough minutia-based comparison. Pre-selection therefore allow to filter a database for the most relevant individuals for every identification query. This reduces the recognition system's workload and enables dealing with larger databases.

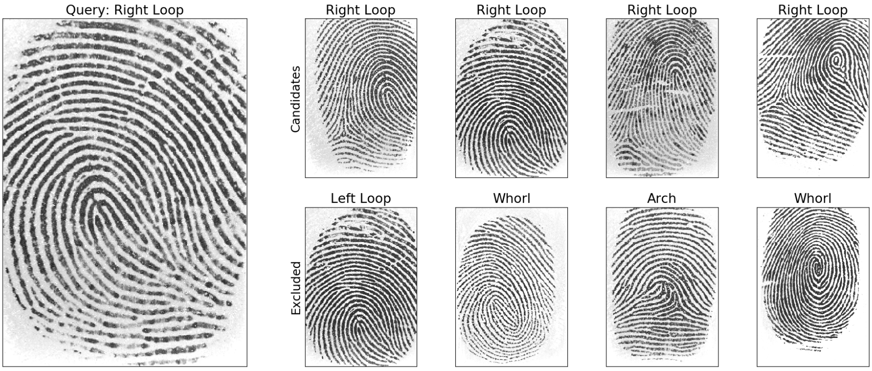


**Figure 13.1:** Some references in a biometric reference database share the same characteristics with the query fingerprint. Those are candidates worth a closer inspection by the thorough minutia-based comparison. Fingerprint Classification uses Henry pattern types as a relevant characteristic, e.g. right loops, left loops, whorls, or arches. By such a pre-selection, Fingerprints not sharing the characteristic with the query item can be excluded from further thorough comparison.

Besides pre-selection algorithms, there are more ways to improve the speed and accuracy of a fingerprint recognition system. Feature extraction as well as biometric recognition can benefit from an *alignment* of the fingerprint samples, i.e. transforming samples to a specific position and orientation. The feature extraction process may benefit from an alignment, e.g. if specialized fingerprint models can be applied, which require an alignment. This may improve mainly on the accuracy of the biometric feature extraction. Biometric comparison may benefit from an alignment even with respect to biometric performance and speed. If aligned samples are compared, the degrees of freedom arising from orientation and position are eliminated. This simplifies the comparison process significantly. Biometric decisions can be made faster and more reliable. In the domain of template protection, there are approaches which even explicitly require an alignment [314].

In the end, there are plenty of processes, which can be improved. For example, fingerprint samples can have very different characteristics, which arise from their data origin, e.g. rolled and plain fingerprint samples. Being able to identify the

| Approach | FIDX-10K | | | FIDX-50K | | |
|---|---|---|---|---|---|---|
| | **ER$_{100}$** | **ER$_{1000}$** | **IS** | **ER$_{100}$** | **ER$_{1000}$** | **IS** |
| MCC | 5.898% | 9.888% | 1.215% | 6.450% | 15.140% | 0.954% |
| *DIMI* | 20.0% | 22.0% | 0.723% | 20.0% | 36.0% | 0.747% |

**Table 13.1:** Results at FVC-ongoing. For definitions of the metrics ER, PR, and IS see Section 1.3.

data origin enables application of dedicated process. This can in turn improve the biometric performance.

## 13.2 State of the Art

### 13.2.1 How to assess approaches efficient processing structures?

The relevant metrics and evaluation for the various aspects of efficient processing structures differ. Evaluation shall focus on the intention of a certain aspect but keep the biometric recognition performance and processing troughput in mind.

Fingerprint Indexing most likely has the highest potential for improving fingerprint recognition systems. There is an independent benchmark at FVC-ongoing for Fingerprint Indexing [68]. This benchmark therefore allows to identify state of the art in Fingerprint Indexing. As FVC-ongoing can be considere most important for evaluation, this work focusses on the provided metrics at this benchmark: PR, ER, and IS (see Section 1.3). FVC-ongoing additionally reports on memory consumption and speed.

There are more sophisticated metrics for reporting aspects of workload reduction [69]. Those metrics exceed the definitions by ISO and FVC-ongoing. Unfortunately, most of these metrics require a baseline algorithm for the thorough comparison. In general, such a baseline is not given for fingerprint recognition. In the special case of FVC-ongoing, thorough comparisons after indexing is not intended at all. Thus, applicability of these metrics is limited in this domain.

About two hundred algorithms or rather versions of algorithms were evaluated at the FVC-ongoing benchmark so far.[1] However, results are reported only for two algorithms: *Minutia Cylinder Code* (MCC)[49] and our *Deep ISO-Minutia Indexing* (DIMI)[263] (see Sections 13.2.2 and 13.2.3). Table 13.1 contains the reported results. The table allows identification of the state of the art for Fingerprint Indexing.

[1] https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/Statistics.aspx

### 13.2.2 Hand-crafted approaches

The most prominent approach in Fingerprint Indexing is MCC [49]. MCC uses fingerprint minutiae as input features for the index vector generation. The idea of MCC is to represent neighbourhoods of minutiae in a compact and fixed-length representation. Neighbouring minutiae are represented by their relative position and angular difference with respect to a central anchor minutia. The representation is encoded on a $T^2$ torus. As a simplification such tori are displayed as cylinders. The actual biometric comparison is reduced to the comparison of such cylinders: Similar cylinders indicate similar minutiae neighbourhoods. There are published results for MCC at FVC-ongoing. MCC can be considered as state of the art for many years. There are many variations of this approach, which improve on several aspects of MCC [303][10][334][11]. The original MCC even comes as a publicly available Software Development Kit (SDK), which allows to integrate it and improve on it.

Besides fingerprint minutiae, there are other features, which can be used for Fingerprint Indexing. There are approaches using orientation fields, ridge textures, or biometric scores. Even hybrid approaches using more than one feature were proposed. As there was no recently published survey, this dissertation will present research review survey providing an overview on the usage of different features used for Fingerprint Indexing in Chapter 14 [256].

### 13.2.3 Learning-based approaches

Up to now, there was no learning based approach. Thus our work presented in Chapter 16 is the first for which results are published in FVC-ongoing This approach is called DIMI [263]. The idea is to use CNNs to learn fixed-length representations for entire fingerprint samples. Like in MCC, DIMI uses neighbourhoods of fingerprint minutiae as input features. DIMI improves significantly on the aspect of IS compared to MCC. DL-based approaches can therefore be considered also as state of the art in Fingerprint Indexing besides MCC.

## 13.3 Contributions

About a hundred approaches were proposed for Fingerprint Indexing so far. A research review survey provides overview on the features used for Fingerprint Indexing (see Chapter 14) [256]. This survey also inspects the quality of the body of research in Fingerprint Indexing.

A new approach using CNNs for Fingerprint Indexing was presented: DIMI (see Chapter 16) [263]. DIMI performs at least competitive to state of the art MCC. It even outperforms MCC in some aspects. It can therefore also be considered state of the art.

Besides Fingerprint Indexing, there are more aspects to improve on in efficient processing structures. An unsupervised learning strategy for CNNs was presented, which allows learning a rotational alignment (see Chapter 15) [264]. It was also shown, that CNNs can be used to estimate the data origin of fingerprint samples (see Chapter 17) [262].

# Chapter 14

# Survey on Features for Fingerprint Indexing

**Summary** About a hundred approaches were proposed for fingerprint indexing so far. This work provides a survey over the multitude of features, which were proposed for fingerprint indexing.
This work was published in the journal *IET Biometrics* in 2018.

[256] P. Schuch. Survey on features for fingerprint indexing. *IET Biometrics*, June 2018. ISSN 2047-4938. URL `http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2017.0279`

**Abstract** Nowadays, several biometric databases already contain millions of entries of individuals. With an increasing number of enrolled individuals, the response time of queries grows and can become critical. Fingerprint Indexing offers a set of techniques to reduce the workload of entries, which have to be compared thoroughly. This work surveys research on such techniques. It focuses on the features of fingerprints, which are used as input. This survey also provides assessment on the quality of the body of research in this field. Deficiencies herein are identified, e.g. there is a lack of common datasets and metrics used for testing.

## 14.1 Introduction

Biometric systems are a widespread means for identification of individuals today. Response times of such systems mainly depend on three aspects: the response time for an individual comparison of two fingerprint samples, the size of the database to be searched, and the actual hardware used for biometric comparison. All three aspects result in the biometric system's throughput.

As biometric databases in general tend to grow over time, the throughput declines over time. Nowadays, there are biometric systems rolled out, which contain several million individuals. The most prominent and largest example is the Aadhaar project in the Republic of India. It already contains more than 100 million entries and targets over one billion people.

In general, classical fingerprint comparison of fingerprint minutiae is computationally expensive. This survey will refer to this exhaustive comparison as the *thorough comparison*. Besides scaling the hardware, there is another way to increase the throughput. While some fingerprints are very similar, some fingerprints are very different. When comparing fingerprints, only those comparisons are worth a closer look, where both fingerprint samples are similar. For many comparisons, one can decide at the first glance that both fingerprint samples are not mated. Thus, one does not need to perform the thorough comparisons on the entire database but only on a subset. This reduces the mean response time for an individual comparison indirectly. This reduction can be done by any kind of filtering the entire dataset for those, which are most likely relevant for the current identification query. The number of thoroughly evaluated entries is therefore reduced. This is usually expressed as the so-called *penetration rate*, i.e. what ratio of the dataset has to be compared thoroughly against. If the combination of time for overhead of prefiltering and time for thorough comparison of the remaining entries reduces the mean response time, the throughput can be increased.

There are mainly two kinds of such prefiltering: *Fingerprint Classification* (FC) and *Fingerprint Indexing* (FI). The former clusters all fingerprints into distinctive classes. The most common clustering assigns one out of five pattern types to a fingerprints. FC can reduce the mean response time. However, it has three drawbacks: The clustering may be ambiguous for many fingerprints, it may fail, if the fingerprint sample contains only a small part of the finger's area, and the penetration rate is not very low due to the small number of classes. FI assigns one or more index values to each fingerprint sample. Usually, a fixed number of features is concatenated to a *fixed-length* feature vector. This in turn allows a rough but computational simple comparison. The comparison is no hard classification like in FC. It is continuous and therefore allows lower penetration rates. FI may therefore be superior to FC. The generation of fixed-length feature values is common also for other biometric traits. Examples are the so-called Iris-Code[62] for iris recognition or Eigen-Faces[288] for face recognition. For both biometric traits such an indexing is distinctive enough to allow not only prefiltering but even identification.

Various approaches for FI have been proposed. This work surveys the research on this topic. It focuses on the features of the fingerprint samples, which are used to generate the index vectors. There is no other survey on the features for FI yet. In addition, the quality of the body of research is assessed.

The rest of this work is organized as follows: Section 14.2 gives a short introduction into FI. The actual survey process is described in Section 14.3. Section 14.4 categorizes and describes the reviewed research items. Section 14.5 gives an overview on the datasets and metrics, which have been used in the surveyed works. A summary with conclusions of this survey can be found in Section 14.6.

## 14.2 Indexing

The technique of FI consists of multiple processing steps. Fingerprint samples are taken as inputs. The final output is a candidate list $\mathcal{C}$.

FI makes use of an index generating function $F$, which maps a fingerprint sample on an identifier. This identifier is called an *index* or an *index vector*. An ideal index would be unique for every fingerprint. Thus, all fingerprint samples of the same fingerprint would be mapped to the same index. In general, the index generating function is not injective, i.e. more than one sample may be assigned to the same index. In addition, two samples of the same fingerprint may be mapped to different index vectors. Thus, the index generating function is not distinctive enough to generate a unique index.

However, the similarity between indices indicates the likelihood for both being mated, i.e. belonging to the same finger. The comparison of indices results in an indexing score $s_i$ representing the likelihood. W.l.o.g. one can assume, that a higher score indicates a higher likelihood. This means that the order of the indexing scores matters. The actual values and relations in magnitude do not necessarily have an explicit meaning.

A requirement for FI is, that the comparison of indices has to be way faster than the typical thorough comparison based on fingerprint minutiae. This aspect of speed is usually achieved by two techniques: comparing index vectors of fixed length and a computationally simple comparison, e.g. $L_2$ norm of the difference between indices.

The result of an FI for a single query is a candidate list $\mathcal{C}$ of identities. This list $\mathcal{C}$ contains a subset of those candidates in the database, which are most likely targets for a given query sample. FI can be interpreted as a fast prefiltering before the thorough comparison will be performed. Figure 14.1 visualizes, how FI can be integrated schematically into the entire biometric comparison workflow. The index generating function $F$ generates an index $F(q)$ for a given query sample $q$. This index $F(q)$ is compared against the set of indices $\{F(\text{id}) : \text{id} \in DB\}$ of all entries id in database $DB$. Each comparison between two indices $F(q)$ and $F(\text{id})$ results in an indexing score $s_i(F(q), F(\text{id}))$. Then an *initial candidate* list $\mathcal{C}$ of identities is generated. If applicable, the initial candidate list is reduced further in an additional *List reduction* process into a *final candidate list*. If no list reduction is applied, the initial candidate list can be passed as the final candidate list. For the sake of simplicity, this survey will only deal with the final candidate list and define it as the candidate list $\mathcal{C}$. Finally, all identities in the candidate list are evaluated by the thorough comparison against the query $q$. The final biometric decision is then carried out only on the remaining candidates. The candidate list $\mathcal{C}$ can be

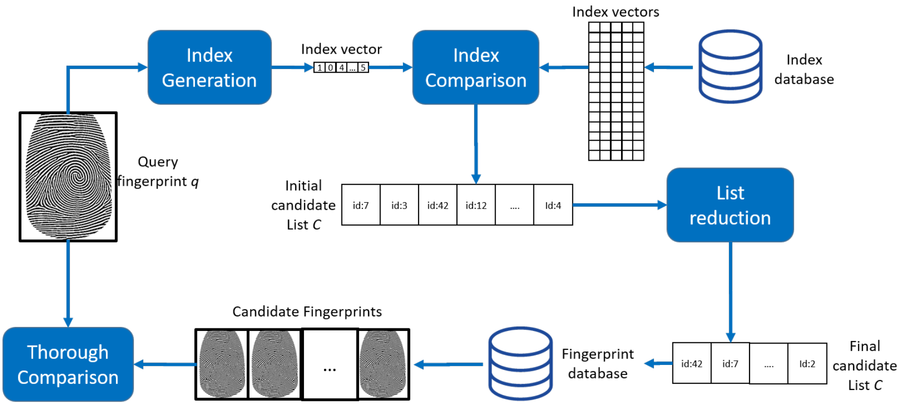


**Figure 14.1:** A fingerprint recognition workflow incorporating Fingerprint Indexing

generated according to several policies. Each represents its own idea of how to use the indexing scores generated by the indexing. Cappelli et al. identified the five typical policies on how to reduce the candidate list $\mathcal{C}$:

- *Fixed threshold*: All indexing scores are compared against a fixed threshold $\theta$. Only those candidates exceeding the threshold $\theta$ are added to the candidate list $\mathcal{C}$. Thus, the candidate list $\mathcal{C}$ is of variable length. In this case the indexing score $s_i$ is interpreted as a probability for being the a mated comparison. Therefore, the fixed threshold $\theta$ represents some kind of probability, which is at least necessary for being a mated comparison.
$\mathcal{C} = \{\text{id} : s_i(F(q), F(\text{id})) > \theta, \text{id} \in DB\}$

- *Top N ranking*: In this scenario the candidate list consists of a fixed number $N$ of candidates, which achieved the highest indexing scores for a single query $q$. The constant length of $\mathcal{C}$ allows a forecast on the runtime of an entire query. Only the order of the index scores is relevant in this case. The actual values of the index scores are irrelevant. This may help in more difficult comparisons, e.g. those in which indexing scores may be low due to bad quality.

- *Variable threshold on score differences*: The highest indexing score $s_{max}(q)$ for a query is identified. The candidate list $\mathcal{C}$ contains all candidates resulting in a comparison score not smaller than the maximum indexing score reduced by a given offset $\delta$. This results in a candidate list $\mathcal{C}$ of variable length.
$\mathcal{C} = \{\text{id} : s(F(q), F(\text{id})) > s_{max}(q) - \delta, \text{id} \in DB\}$

- *Variable threshold on score ratio*: This approach is similar to the approach *Variable threshold on score differences*. The highest indexing score $s_{max}(q)$ for a query is identified. The list $\mathcal{C}$ contains all candidates resulting in an index score exceeding a given ratio $\rho$ with respect to the highest found index score. This also results in a candidate list $\mathcal{C}$ of variable length.
$\mathcal{C} = \{\text{id} : s(F(q), F(\text{id}))/s_{max}(q) > \rho, \text{id} \in DB\}$

- *Oracle*: In this case, for every query the candidate list is of exactly the optimal size, which is required to include the correct identity. The workflow would be to thoroughly compare the candidates in the candidate list until the correct one is identified. This implies a perfect thorough comparison. Therefore, this policy is mainly of theoretical value.

Generating the candidate list can be quite time consuming, e.g. if a very long lists of indexing scores needs to be sorted. In such cases, the candidate list generation can take a large amount of the total time of the FI. The policy for the candidate list construction might therefore be considered as a critical component. All policies have their own reasoning and are reasonable to some degree. None of the policies is superior to the others by design. Every policy has its own to reasonable metrics.

Other forms of candidate list reduction are possible. For example, if one was able to decide or knew whether a fingerprint sample belongs to a male or a female, one could reduce the candidate list by gender. Dantcheva et al. provided a comprehensive review on such approaches [61].

## 14.3 Survey

### 14.3.1 Survey Process

I decided to search for relevant works in the four most relevant archives in computer science which require a review: *IEEE Xplore*, *ScienceDirect* from Elsevier, *SpringerLink*, and *ACM Digital Library* (ACM). Table 14.1 reports the exact search phrases. Only original research works have been included in the survey. Therefore, some additional filtering had to be applied on the search results. The table also lists the numbers of research items found by the review.

**Table 14.1:** Number of relevant research items after the distinctive review stages and instructions for a reproducible review search.

| Archive | # Initial Results | #After 1$^{st}$ screen | #After 2$^{nd}$ screen | Search Command | Additional Filters |
|---|---|---|---|---|---|
| IEEE Xplore | 139 | 40 | 31 | `(((Fingerprint) AND Indexing) AND Biometric*)` | "Conference Publications Journals" or "Magazines" |
| ScienceDirect | 52 | 21 | 16 | `("Fingerprint Indexing") and Biometric*` | "Journals" |
| SpringerLink | 33 | 15 | 11 | `Biometric AND "Fingerprint Indexing"` | "Conference Paper" or "Article" |
| ACM | 128 | 13 | 5 | `(+Fingerprint +Biometric Indexing)` | *no filter* |
| From reference lists | n/a | n/a | 19 | n/a | n/a |

The first screening was to sort out roughly the irrelevant works from all found results. This was done considering only publication titles and abstracts. Mainly, there were two criteria for inclusion. First, a proposed approach had to be applicable to fingerprint samples. Second, it needed to generate a fixed-length index vector for samples or features, which could be used for FI. In case of doubt, research was kept for closer inspection during the second screening.

During the second screening, the remaining publications were analysed thoroughly. Some works, which had not been sorted out before, were filtered by this stage. No further criteria have been applied in this stage. Deficiencies in the quality of work have been captured by using a survey questionnaire (see Subsection 14.3.2.)

The reference lists of the most popular publications found during the regular survey have also been inspected. This allowed to identify additional relevant works, which had been missed by the search in the archives. Actually, this review is not complete in respect to all publications ever done in the domain of FI. The survey is limited by the described review process. To the best of my knowledge, no ground-breaking features for FI were missed by this review process.

### 14.3.2 Survey questionaire

The approaches found during the survey were evaluated according to a defined catalogue of aspects. Those aspects are:

- **[Level]**: *What is the level of detail?*
  This aspect shall estimate how well the actual methods are described. This is very important for reproduction of the claimed results. The clearer and the more detailed the description is, the higher the chance of reproducibility. However, this aspect is subject to the opinion of the survey's author. This aspect's ratings range from a good level (🟢), over a fair level (🟡) to a bad level (🔴) of description.

- **[Repr]**: *Is the approach working in a local or a global manner?*
  Some approaches inspect *local* structures, e.g. neighbourhoods of fingerprint minutiae. Others use a *global* representation, e.g. the orientation field of the entire fingerprint sample.

- **[Mod]**: *Which biometric modality is addressed?*
  This aspect indicates, whether an approach is bound to fingerprints (FP) only. Some approaches are also proposed to be applicable for the modality *Palm* (P) and some even to any modality (*).

- **[Multi]**: *Is a solution provided to process more than one finger at a time?*
  Some biometric systems use more than one fingerprint as an identifier, e.g. all four fingerprints of a hand. Checkmarks indicate, whether the approach provides a solution for the combination of more than one fingerprint.

- **[Index]**: *Is a single index generated for sample or are there multiple indices for each feature?*
  In general, a single index vector per sample allows a simpler comparison

workflow, because in the other case special consolidations on the sub results have to be performed. This aspect correlates slightly with the aspect of *Repr*: most local approaches generate indices per feature and most global approaches generate indices per sample. However, there are counter-examples.

- **[$\mathcal{O}(\cdot)$]**: *Is there any assessment on computational complexity?*
  FI is meant to improve a system's throughput. A proposed approach shall therefore be evaluated on the effort which must be performed to apply the approach. An approach is even useless, if it is so computationally expensive that there is no benefit in terms of the throughput at the end. Checkmarks indicate any assessment of the aspects.

- **[⏱]**: *Is there any assessment on time consumption?*
  The aspects of computational complexity $\mathcal{O}(\cdot)$ and timing ⏱strongly depend on each other. Nonetheless, the former is the more significant one. Checkmarks indicate any assessment of this aspect.

- **[Cit.]**: *How often was the work cited?*
  The number of citations can be interpreted as a rough indicator of the impact of an approach. The actual number of citations was measured on Google Scholar on November 1st 2017. The higher the count of citations, the more important a publication can be assumed. Google Scholar includes self-citations into the citation count. Self-citations distort this indicator.

An additional aspect would be the memory consumption of an approach. In general, this aspect is not addressed by the approaches. But it can be derived by the reader.

## 14.4 Relevant approaches

The found approaches can be grouped into four domains of features: fingerprint minutiae, fingerprint ridges, orientation fields, and biometric scores. Each domain will be described in its own subsection in the following. Tables 14.2-14.6 give an overview over the found approaches with respect to the aspects monitored by the survey questionnaire.

### 14.4.1 Approaches using Fingerprint Minutiae

The vast majority of approaches for FI uses fingerprint minutiae (see Figure 14.3) as a feature. Due to the use of minutiae, all approaches in this domain are bound to the biometric modality of fingerprints. Fingerprint minutiae are also the most common features used for thorough comparison of fingerprint samples.

Minutiae are well suited even for biometric identification. Thus, approaches using fingerprint minutiae for FI make use of very descriptive and powerful features. Another advantage over other features is the fact, that most of these approaches are applicable without any knowledge about the original fingerprint sample. They can therefore be also applied to already deployed systems, in which no fingerprint images are available.

**Table 14.2:** Approaches making use of minutiae.

| Authors | Ref. | Approach | Level | Repr | Mod | Multi | Index | $\mathcal{O}(\cdot)$ | ⏱ | Cit. |
|---|---|---|---|---|---|---|---|---|---|---|
| Bai et al. | [9] | k-nearest neighbours | 🟢 | local | FP | | feature | | ✓ | 6 |
| Bai et al. | [10] | Statistics on MCC | 🟢 | local | FP | | feature | | | 0 |
| Bebis et al. | [21] | Delaunay | 🟢 | local | FP | | feature | ✓ | | 189 |
| Benhammadi et al. | [23] | Minutia Code | 🟢 | local | FP | | feature | | | 68 |
| Bhanu et al. | [25] | Triplets | 🟢 | local | FP | | feature | | ✓ | 223 |
| Biswas et al. | [27] | Triplets | 🟡 | local | FP | | feature | | | 24 |
| Cappelli et al. | [52] | MCC | 🟢 | local | FP | | feature | | ✓ | 111 |
| Chen et al. | [56] | Matching of minutia subsets | 🟢 | local | FP | | n/a | ✓ | ✓ | 28 |
| Germain et al. | [85] | Triplets | 🟢 | local | FP | | feature | | ✓ | 287 |
| Gago-Alonso et al. | [79] | Delaunay triangulation | 🟢 | local | FP | | sample | | | 25 |
| Hartloff et al. | [106] | Sequences of minutiae | 🟢 | local | FP | | feature | | ✓ | 13 |
| Iloanusi et al. | [114] | Quadruplets | 🟢 | local | FP | ✓ | feature | | ✓ | 22 |
| Iqbal et al. | [117] | Triplets and quadruplets | 🟢 | local | FP | | sample | | ✓ | 1 |
| Jain et al. | [132] | Geometric representation | 🟢 | global | FP | | sample | ✓ | | 1 |
| Jain et al. | [131] | Geometric representation | 🟢 | global | FP | | feature | | | 2 |
| Jayaraman et al. | [138] | Minutia in relation to cores | 🟢 | global | FP | | sample | ✓ | | 15 |
| Khachai et al. | [150] | Delaunay triangulation | 🟢 | local | FP | | feature | | | 3 |
| Khodadoust et al. | [156] | Expanded Delaunay triangulation | 🟢 | local | FP | | feature | | ✓ | 0 |
| Khodadoust et al. | [155] | Triplets of minutiae and cores | 🟢 | local | FP | | feature | | | 1 |
| Kovacs-Vajna | [161] | Triplets | 🟢 | local | FP | | feature | | ✓ | 419 |
| Kumar et al. | [166] | Nearest neighbours | 🔴 | local | FP | | sample | | | 2 |
| Le et al. | [169] | Triplets | 🟢 | local | FP | | feature | ✓ | | 2 |
| Le et al. | [170] | Hashing on neighbourhoods | 🟢 | local | FP | | feature | ✓ | | 0 |
| Li et al. | [178] | Minutia disks | 🟢 | local | FP | | feature | | | 6 |
| Li et al. | [78] | Minutia disks | 🟢 | local | FP | | sample | | | 0 |
| Liang et al. | [182] | Delaunay with minutia type | 🟢 | local | FP | | feature | ✓ | | 38 |
| Liang et al. | [183] | Delaunay with minutia type | 🟢 | local | FP | | feature | ✓ | | 96 |
| Liu et al. | [185] | Minutiae and BioCode | 🟢 | global | FP | | sample | | | 23 |
| Mansukhani et al. | [204] | Trees of neighboorhoods | 🟢 | local | FP | | feature | | ✓ | 21 |
| Muñoz et al. | [211] | Low order Delaunay triangles | 🟢 | local | FP | | feature | | | 2 |
| Muñoz et al. | [212] | Delaunay triangulation | 🟡 | local | FP | | feature | | | 2 |
| Muñoz et al. | [213] | Extended Delaunay Triangulation | 🟢 | local | FP | | feature | | ✓ | 0 |
| Nagati | [216] | Minutiae around cores | 🟢 | local | FP | | sample | | | 0 |
| Reddy et al. | [241] | Triplets | 🟢 | local | FP | | feature | | | 1 |
| Ross et al. | [245] | Tiplets with ridge curves | 🟢 | local | FP | | feature | | | 50 |
| Vandana et al. | [296] | Lower order Delaunay triangles | 🟢 | local | FP | | feature | | | 8 |
| Vij et al. | [297] | Selection of quadruplets | 🟢 | local | FP | | feature | | ✓ | 7 |
| Wang et al. | [302] | Speheric variation of LSH | 🟢 | global | FP | | sample | | ✓ | 1 |
| Wang et al. | [303] | Shrinking of MCC | 🟢 | local | FP | | sample | ✓ | | 9 |
| Xu et al. | [312] | Spectral representation | 🟢 | global | FP | ✓ | sample | | | 2 |
| Yang et al. | [315] | Pixel Look-up via 3 minutiae | 🟡 | global | FP | | sample | | | 0 |
| Zhou et al. | [333] | Triplets | 🟢 | local | FP | | feature | | ✓ | 8 |
| Zhou et al. | [334] | Alternative hasing for MCC | 🟢 | local | FP | | feature | | | 0 |

A set of approaches is working on sets of few minutiae. For each set features are calculated, which describe the relation of the minutiae. Germain et al. were the first to propose triplets (see Figure 14.3(b)) of fingerprint minutiae [85]. The triplets can then be represented by their geometric features, e.g. angles and side lengths. The number of possible combinations of minutiae to triplets is very large. When no restrictions on the triplets apply and an index is generated for every triplet, the number of triplet comparisons will be even larger. Therefore, restrictions seem reasonable. Kovács-Vajna, Bahnu et al., Reddy et al. and Zhou et al. proposed variations in the selection of relevant triplets and features to be extracted [161][25][241][333]. These strategies allow to keep the number of triplets to a reasonable order. Ross et al. enriched triplet features with the ridge curves associated with the vertices [245]. The information on adjacent curves is of course a valuable information. By the way, this approach is quite similar to the methods human examiners would apply. Biswas et al. extended the features of triplets with information on the local *curvature* of the ridge structure [27]. Each region in a fingerprint sample has its own characteristic curvature. Thus, adding curvature information significantly increases the total information of a single minutiae. The gain in information of course increases by using not one but three minutiae. Triplets become very descriptive in regions of strongly varying curvature. Khodadoust et al. proposed to use triplets containing two minutiae and a singularity as corners [155]. This approach of course depends on a reliable detection of the fingerprint singularities. Detection of the singularities is challenging especially for fingerprint samples of bad quality. Singularities may not be detected at all in partial fingerprints or in fingerprints with pattern types without singularities at all, i.e. *arches*.

Several approaches proposed to use a Delaunay triangulation for selection of relevant triplets. Sampling triplets in such a manner is a special and popular strategy in triplet selection. Bebis et al. were the first to use Delaunay triangulation [21]. The advantage of using this sampling strategy is that a Delaunay triangulation generates unique sets of triplets. Even though the Delaunay triangulation may slightly differ, when the minutia positions are disturbed, most of the Delaunay triangulation usually works stable for most of the entire fingerprint sample. Liang et al. proposed to variant integrating the minutia type as a feature [182][183]. The type of a minutia can either be a so-called ridge *ending* or a ridge *bifurcation*. Adding such information to the description of minutiae can enhance the information significantly. However, distinguishing both types is challenging especially for fingerprint samples of low quality. Vandana et al. proposed to use triplets from a lower-order Delaunay triangulation [296]. This approach increases the amount of generated triplets. The approach gets more tolerant to spurious and missing minutiae. Muñoz-Briseño et al. additionally used distances to the next nearby singularity as a feature [212]. This information can be interpreted as a rough indicator for the region in the fingerprint sample. Again, detection of singularities can be challenging or even impossible. They further proposed two variations of the triplet selection by the Delaunay triangulation [213][211]. Khodadoust et al., Khachai et al., and Gago-Alonso also proposed further variations of the Delaunay triangulation [156][150][79]. Those approaches mainly deal with the challenges arising due to missing and spurious minutiae or from positional variations of the minutiae.

Some approaches use more than three minutiae. The more minutiae one uses, the more informative the set of minutiae becomes. These approaches suffer from missing and spurious minutiae, too. Vij et al. and Iloanusi et al. proposed to use quadruplets of fingerprint minutiae [297][114]. In some cases of biometric comparisons, two fingerprint samples from the same fingerprint share only a small area. In such cases, finding shared quadruplets will be harder than finding shared triplets. Thus, the strong gain in information comes with drawbacks for mated comparisons with small overlapping area. Iqbal et al. proposed to combine triplets and quadruplets of minutiae and perform cascaded filtering on these [117]. This approach therefore tries to combine the benefits from triplets and quadruplets. Cappelli et al. proposed to represent minutiae and their relative neighbourhood in a compact form called *Minutia Cylinder Code* (MCC) [52]. In this approach a cylinder describes the relative position and relative angles of the neighbouring minutiae (see Figure 14.2). The cylinders represent a special tessellation grid. Neighbouring minutiae are represented as Gaussians on the grid. This deals with the sampling errors on the grid and also enables tolerance to slight positioning errors of minutiae. Each plane in the cylinder represents a relative angle with respect to the central minutia. By the way, actually it is not a cylinder but a torus, which allows to deal with the cyclicity of angles. MCC is the only approach, which was evaluated at benchmark *FVC-ongoing* [68]. FVC-ongoing is the only available independent benchmark

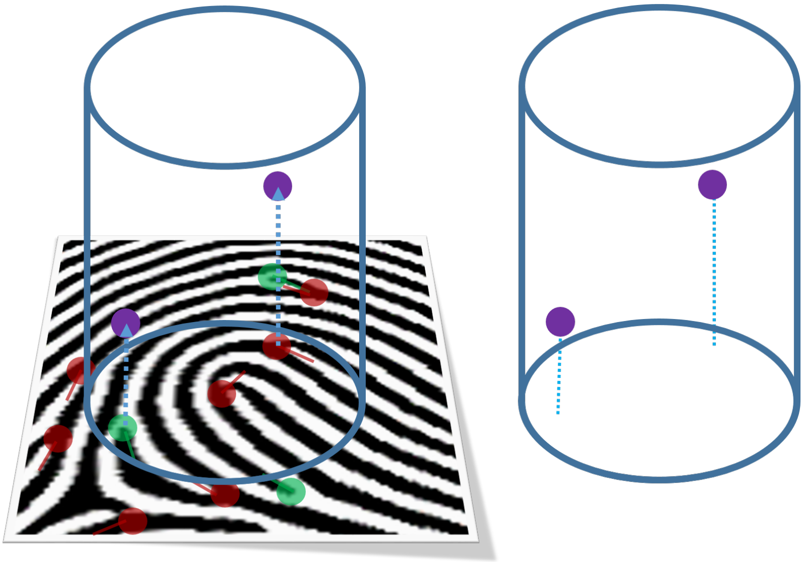


**Figure 14.2:** Minutia Cylinder Code encodes a neighbourhood of minutiae. Each neighbouring minutia (purple) is represented with its relative angle and position to a central minutia

for FI. There is an SDK for MCC available by the way. MCC is the base for some variations. Wang et al. proposed to reduce the size of the cylinders [303]. Even though the original MCC has a compact format, only a small share of each cylinder is non-zero, i.e. the points where the neighbouring minutiae lie. This fact allows reduction of the size. The original MMC also has a straightforward tessellation grid. Sampling and quantisation leaves space for improvement here. Bai et al. proposed therefore to use statistics on the cylinders for an improved quantization of the cylinders [10]. Zhou et al. proposed an alternative hashing for MCC [334]. Li et al. proposed two variants of descriptions of neighbourhoods, described as *Minutiae Disks* [178][78]. Neighbouring minutiae are represented here in Polar coordinates. Polar coordinates enable a more natural dealing with positional and angular relations compared to Cartesian coordinates.

There are further approaches describing local neighbourhoods of minutiae (see Figure 14.3(a)). Those approaches usually use the nearest neighbouring minutiae around a central minutia. These neighbourhoods therefore are local descriptions. Such descriptions can be interpreted as small puzzle pieces. Those usually allow to deal with biometric comparison of fingerprint samples with small overlap. In such cases only very few puzzle pieces match between the fingerprint samples. Kumar et al. proposed to add some undefined features to minutiae [166]. Mansukhnai et al. and Bai et al. proposed to use trees for fast comparison of local minutia neighbourhoods [204][9]. Hartloff et al. used series of neighbouring minutiae and concatenated them to sequences (see Figure 14.3(c)) of minutiae [106]. Those sequences have similar features like triplets or quadruplets. The concatenation of minutiae to sequences can be interpreted as a sampling strategy of neighbours like in the approaches using triplets. However, this approach might be promising for comparisons of mated fingerprints with very small overlaps. This interpretation of minutia sequences as *Strings* allows to use methods from string processing. Benhammadi et al. proposed to describe the minutia by their surrounding local orientations in so-called *Minutia Code* [23]. Local orientations give an idea, in which region of a fingerprint the sampled minutia is in. The representation is especially expressive in regions of high curvature or near orientation field singularities.

Some approaches deal in more detail with the index generation from the features. Le et al. improved the index generation [169][170], e.g. with error correcting codes. Wang et al. proposed a variation of Locality-sensitive hashing (LSH) on fingerprint minutiae [302].

There are also ideas, which only aim on a fast comparison, but do not necessarily use a fixed-length representation. For example, Nagati proposed to use only the minutiae in the central region for a fast but thorough comparison [216]. This region usually contains the highest curvature and dynamic. One can assume, that most of the information is clustered in this region. However, by doing so available information is discarded. The reduction to a quite small region automatically introduces the problems arising from small overlap between fingerprint samples. Jayaraman et al. described the local neighbourhoods in the regions around the singularities [138]. Chen et al. proposed to compare just a subset of minutiae in a thorough manner [56]. Both approaches suffer from the same drawbacks as the approach proposed by Nagati.

Other approaches use minutiae for a global representation. Jain et al. proposed two variations of a geometric representation of all minutiae of a fingerprint sample [132][131]. This representation is called *Spiral Tree* and can be used for feature extraction. The generated representation looks like a snail shell. This representation is quite helpful for visual inspection, since it allows easy comparisons between two fingerprint samples. Weaknesses of the approach might be the detection of the inner most point of Spiral Tree and the disturbance from spurious and missing minutiae. Yang et al. proposed to assign a value from a random look-up table and three pixels each to each pixel in an image [315]. The random look-up induces revocability. Xu et al. proposed to use a complex representation of the fingerprint minutiae [312]. Each minutia is represented by a complex impulse in the image domain. A minutia's orientation is encoded into the phase of the impulse. By applying Fourier transformation and mapping to Polar coordinates this approach is invariant to translation and rotation. Fourier transformation enables the invariance to translations. Rotations in the image domain result in translations in the Polar coordinates of the Fourier spectrum. These translations can be dealt with by application of correlation between two samples. Thus, this approach nicely uses signal processing techniques. Liu et al. proposed to combine minutiae with a modification of *BioCode* (which originally included an additional token) [185]. External tokens induce extra effort for users but usually enable them to apply verification approaches rather than identification approaches, of course.

### 14.4.2 Approaches using Fingerprint Ridges

Several approaches for FI make use of the ridges of the fingerprint samples. Thus, those methods are dealing with the textures in an image. Such approaches usually assume that fingerprints actually look similar between multiple impressions. This assumption does only hold for some samples. Usually, two fingerprint samples of the very same fingerprint vary. The stronger the variation is, the less applicable those approaches are. Table 14.3 lists the found approaches. Four approaches propose to use local descriptors, which are commonly used for dealing with textures. Those techniques are often used for the task of image registration, which can be seen as a similar task to finding out how two fingerprint samples fit together. Shuai et al. proposed to use a fixed number of *Scale-invariant feature transform* features

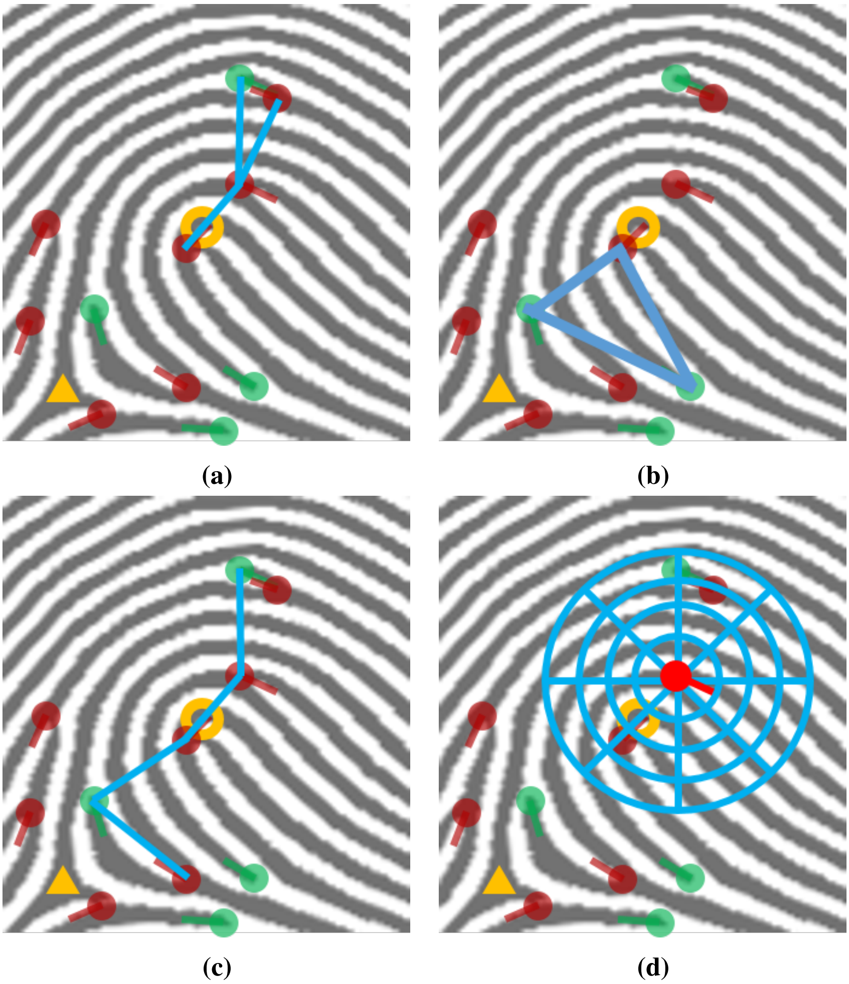


**Figure 14.3:** Minutia are characteristic points on the ridge structure: endings (green) and bifurcations (red). Other characteristic points are the singularities: delta (orange triangle) and core (orange circle). Minutiae can be used for Fingerprint Indexing in various representations. Usually, neighbouring minutia are described in groups, e.g. nearest neighbours (14.3(a)), triplets (14.3(b)) or a sequence (14.3(c)). In some cases the neighbourhood is sampled at a tessellation grid (14.3(d)).

**Table 14.3:** Approaches making use of ridges/texture.

| Authors | Ref. | Approach | Level | Repr | Mod | Multi | Index | $\mathcal{O}(\cdot)$ | ⏱ | Cit. |
|---|---|---|---|---|---|---|---|---|---|---|
| Feng et al. | [76] | Ridge Invariants | 🟡 | local | FP | | feature | | | 41 |
| He et al. | [108] | SURF and DAISY features | 🟢 | global | * | | sample | | | 15 |
| Komal et al. | [160] | Radon transformation of core region | 🟢 | local | FP | | sample | | | 0 |
| Jakubowski et al. | [137] | Ridge crossing over random lines | 🟢 | global | FP | | sample | | | 3 |
| Jazzar et al. | [139] | Zernike moments | 🟢 | global | FP/P | ✓ | sample | | ✓ | 13 |
| Liu et al. | [191] | Region around singularities | 🟡 | global | FP | | sample | | | 53 |
| Montoya et al. | [326] | Wavelets for core regions | 🟢 | local | FP | | sample | | | 45 |
| Nanni et al. | [218] | Various block-wise local descriptors | 🟢 | local | FP | | feature | | | 57 |
| Shuai et al. | [268] | Subset of SIFT of 3 impressions | 🟢 | global | FP | | sample | | | 67 |
| Yang et al. | [318] | Invariant Moments | 🟢 | local | FP | | sample | | ✓ | 97 |
| Zheng et al. | [332] | SURF features | 🟢 | local | FP | | sample | | ✓ | 4 |

(SIFT)[195] features as a descriptor (see Figure 14.4)[268]. SIFT features are scale-invariant and were quite popular for matching two images, e.g. in computer vision with two cameras. Since actual sizes of structures in a fingerprint are known due to known image resolutions and physiological limitations of fingerprints, the scale-invariance of the SIFT features might be an unnecessary restriction of the feature description. For instance, a twice as large copy of a fingerprint sample would match the original fingerprint perfectly, while two samples of such scaling difference could never be originated from the same fingerprint. Fingerprint samples of low quality usually differ strongly. Such variations make SIFT even less applicable. He et al. proposed a combination of *Speeded Up Robust Features* (SURF)[19] and DAISY[285] features to describe the textures [108]. SURF features are similar to SIFT features, while replacement of internal filters leads to a speed-up compared to SIFT features. The same restrictions as for SIFT features apply for SURF features as well. The speed up is mainly achieved during the feature index generation, in which timing is of little relevance. Zheng et al. used only SURF features, but combined it with clustering [332]. Nanni et al. proposed to use a selection of local descriptors for FI [218]. Some other approaches focus on the description of a small region of the fingerprint sample. Like in the approaches for fingerprints, the reduction to sub region is always discards information. Challenges from comparisons of samples with small overlap do also apply here. Those approaches usually detect the fingerprint singularities first and process only the area around those singularities. Liu et al. were the first to describe this central region [191]. Montoya et al. proposed to use a wavelet decomposition to describe this region [326]. Representation with wavelets usually require a precise alignment of the input samples. Alignment is a research field on its own, which brings its own challenges. Wavelet approaches usually are also disturbed strongly by elastic transformations. Komal et al. described the central region with a variation of the *Radon transformation* [160]. The Radon transformation can be understood as a representation for lines. This makes it an appropriate means for the descrip-

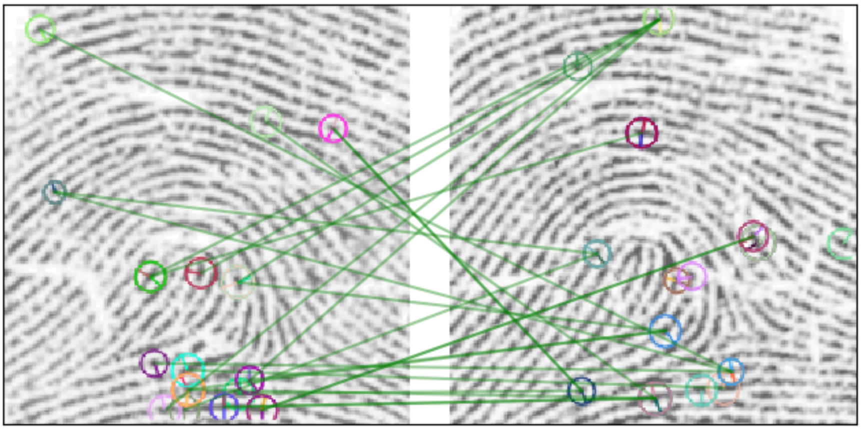


**Figure 14.4:** Scale-invariant features can describe textures (coloured circles). These descriptors can be matched (green lines) and may also be used for FI.

tion of fingerprint ridges. However, the flow of fingerprint ridges is disturbed by fingerprint minutiae. Yang et al. proposed the usage of *invariant* moments in the central region [318]. These moments are invariant to position, scale, and rotation. This makes a precise alignment irrelevant.

Three other texture-based approaches have been proposed. Feng et al. proposed to calculate so-called *Ridge invariants* from the ridge structures [76]. This approach is related to the minutia-based approaches as it describes ridges by minutiae on them. However, the focus here is on the ridges. Indices are generated for lines, which cross over fingerprint ridges. The ridges are identified by minutiae on them. This approach imitates the method a human examiner might use, when he counts ridges between minutiae. The approach therefore also might deal well with elastic distortions of the fingerprint, since those disturbances cannot change the connections of ridges between minutiae. Jakubowski et al. proposed to use the count of crossings of the ridge structure over several random lines [137]. This method does not only describe the texture, but also the orientations found in the fingerprint samples. This approach would require an alignment. Otherwise, the approach would fail, since the random lines would cross different ridges. This approach is not tolerant to elastic deformations. Jazzar et al. proposed to use *Zernike moments* for the description of an entire fingerprint sample [139]. Zernike moments are invariant to rotation by definition and can be modified to be also invariant to scaling and translation. Those moments are derived from a set of complex polynomials, which form an orthogonal base. This results in a compact description of the fingerprint ridge structure.

### 14.4.3 Approaches using Scores

All approaches in this domain of features make use of the conventional and thorough biometric comparison. Those approaches use the biometric comparison scores as a biometric feature on its own. The key to still achieve an improvement in throughput is to compare only a small fraction of the entire search database. The most charming aspect in such approaches is the fact, that one incorporates the same technology as in the thorough comparison. What cannot be compared correctly during FI, will most likely not be compared correctly during the thorough comparison anyway. As the thorough comparison usually makes use of fingerprint minutiae, these approaches can also be linked to the minutia-based approaches in Section 14.4.1. They make use and benefit indirectly of these descriptive features. Improvements over time made in the thorough comparisons, may also result in improvements in such FI approaches. On the other hand, all approaches will suffer from the same challenges, which are also present during the thorough comparison. Most challenging here are partial fingerprints, which potentially result in mated comparisons with small overlap. Table 14.4 gives an overview over the five approaches in this feature domain.

Gyaourova et al. proposed a total of three approaches, which are all quite similar in their basic idea to use a *Bag-of* approach [100][101][102]. The idea here is to identify a set of samples in the search database, which represents some prototypes of fingerprints. All queries are compared only against this set (bag) of prototypical fingerprints. Each score is a single feature in the index (see Figure 14.5). It is therefore a representation by means of the similarity to prototypes. It is assumed that all impressions of the same fingerprint will result in similar biometric scores when compared against the prototypes. This approach is slightly related to the so-called *Doddington Zoo* [67], because it assumes that each fingerprint will generate its individual biometric score distribution. The fix number of thorough comparisons allows assumptions on the processing time. Partial fingerprints may result in small biometric scores for all pattern types. Depending on the policy for candidate list construction, partial fingerprints will then result in very long candidate lists or the correct identity might not be in the list at all. Cappelli et al. proposed to perform only a few comparisons [51]. The idea of Doddington's zoo also applies in this case. For instance, there are some fingerprints matching well with many others. These a called chameleons. For so-called *ghost* fingerprints mated and non-mated comparisons both result in low biometric scores. Murakami et al. proposed a slightly related approach [214]. They used a few scores to imitate a search against an entire database. This approach strongly depends on the predictive power of those few scores.

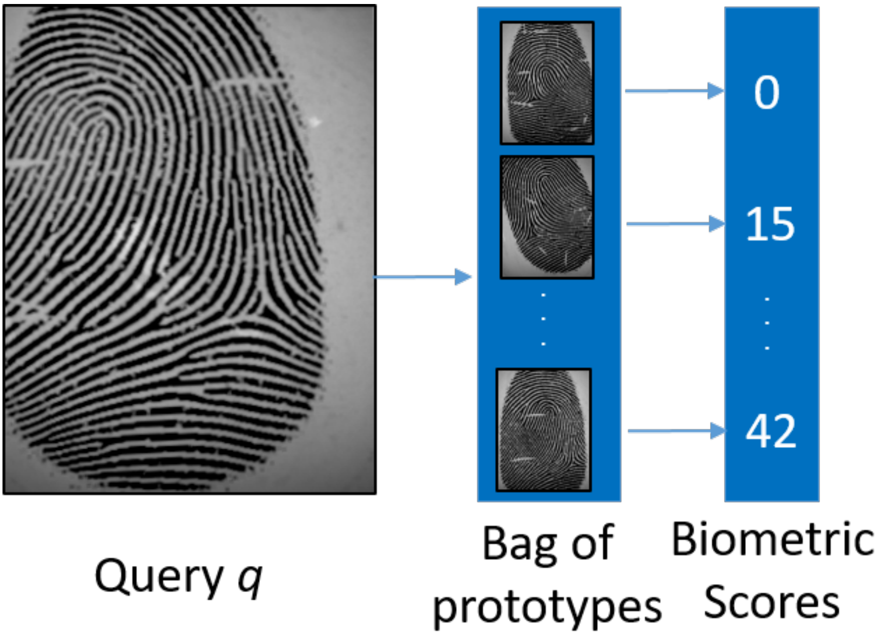


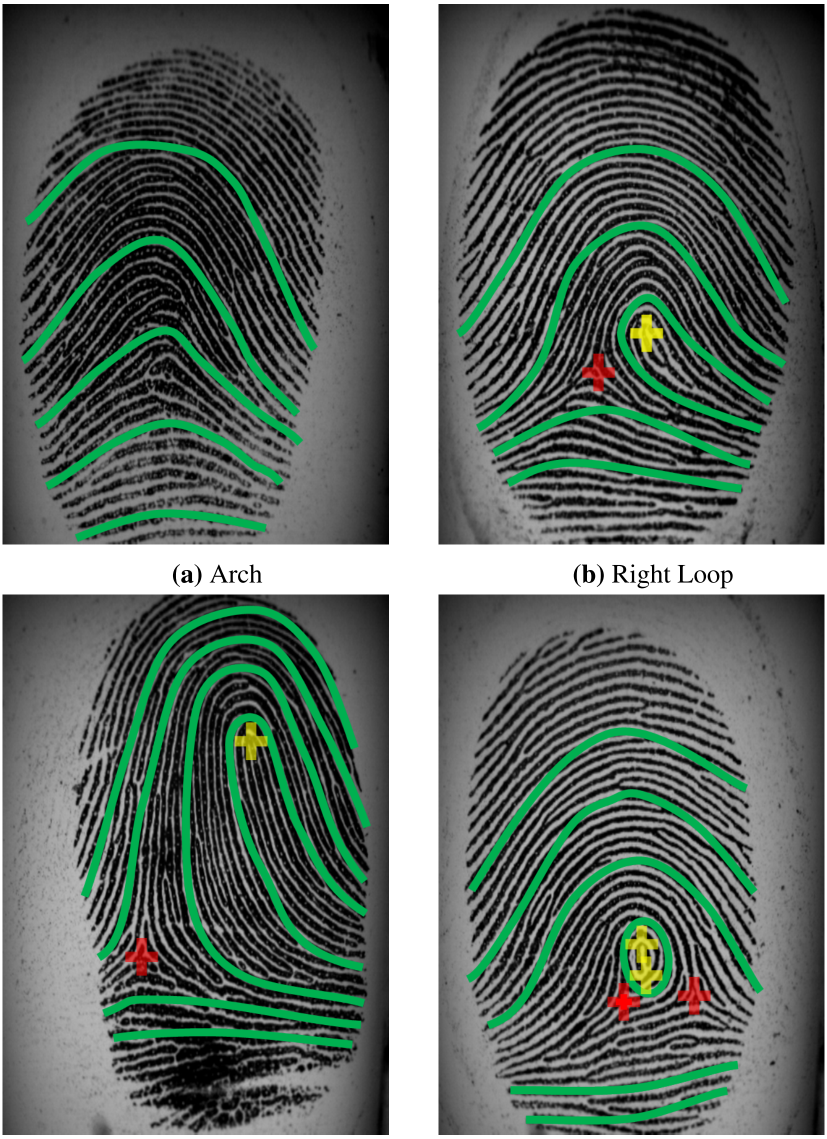
**Figure 14.5:** Biometric scores can be used for Fingerprint Indexing. Comparison against a set of prototypes will result is similar biometric scores for similar fingerprints.

**Table 14.4:** Approaches making use of biometric comparison scores.

| Authors | Ref. | Approach | Level | Repr | Mod | Multi | Index | $\mathcal{O}(\cdot)$ | ⏱ | Cit. |
|---|---|---|---|---|---|---|---|---|---|---|
| Gyaouorova et al. | [100] | Bag of templates | ○ | global | * | | sample | ✓ | | 44 |
| Gyaouorova et al. | [101] | Bag of templates | ○ | global | * | | sample | | | 25 |
| Gyaouorova et al. | [102] | Bag of templates | ○ | global | * | ✓ | sample | | ✓ | 50 |
| Murakami et al. | [214] | Imitation of query | ○ | global | * | | sample | | ✓ | 12 |
| Cappelli et al. | [51] | Evaluation on few scores | ○ | global | * | | sample | | ✓ | 16 |

### 14.4.4 Approaches using Orientation Fields

There are also approaches, which make use of the orientation field of the fingerprint samples. The orientation field is a representation for the local orientations of the ridge structure. It is a so-called 1$^{st}$ level feature. The orientation field is such a distinctive feature that it can be used especially for exclusion during fingerprint comparison (see Figure 14.6). However, estimation of the orientation field is no trivial task. The estimation may fail especially in regions of bad quality. Estimation of the orientation field is a research field on its own [203]. Moreover, orientation fields are also the most frequent used feature in FC approaches [80]. Approaches using the orientation field therefore have similar advantages and drawback as approaches in FC. The main advantage over FC is the fact that there is



**(a)** Arch

**(b)** Right Loop

**(c)** Right Loop

**(d)** Whorl

**Figure 14.6:** An orientation field describes the local orientations (roughly indicated as green lines) of the ridge structure. The orientation fields are dominated by the positional relation and presence of singularities (yellow and red crosses). Orientation fields have already been used for Fingerprint Classification, which classified each fingerprint into a distinctive pattern type class. Each pattern type represents a large variation of orientation fields (compare Figures 14.6(b) and 14.6(c)). Orientation fields can also be used for Fingerprint Indexing.

no hard classification in pattern types. FI allows continuous index vectors. This mainly allows to deal with two challenges in FC. First, even though the classification sorts into single classes, the classification is unambitious in some cases. There is small inter-class variance for some classes of pattern types. The transition from one pattern type to another is continuous in those cases. Second, two fingerprints from the same pattern type may still have really different orientation fields. Actually, there is quite large intra-class variance for some fingerprint classes. For instance, singularities may be quite close to each other in one fingerprint and far away in the other. Descriptive power is not used, if one uses only hard classes like in FC. Another advantage of using orientation fields for FI is the fact that orientations fields are smooth and can be modelled mathematically. This allows to make estimations of what the orientation field looks like in regions close to the area of the actual fingerprint sample. This in turn allows to deal with biometric comparisons with small overlap. However, this hypothetical advantage is not used by any approach. Another charming aspect of using orientation fields is the fact, that the orientation field is visually perceivable and understandable for humans. One can easily see and tell, why FI using orientation fields works in some cases and fails in the other cases. Many approaches are neither invariant to rotation nor to translation. Therefore, those approaches rely on some kind of alignment before processing. All approaches use a global representation and generate a single index for each sample.

**Table 14.5:** Approaches making use of the orientation field.

| Authors | Ref. | Approach | Level | Repr | Mod | Multi | Index | $\mathcal{O}(\cdot)$ | ⏱ | Cit. |
|---|---|---|---|---|---|---|---|---|---|---|
| Jain et al. | [135] | FingerCode | 🟢 | global | FP |  | sample |  |  | 1393 |
| Kavati et al. | [147] | 8 Gabor filters | 🟢 | global | FP |  | sample |  | ✓ | 3 |
| Leung et al. | [176] | Variation of FingerCode | 🟢 | global | FP |  | sample | ✓ | ✓ | 31 |
| Li et al. | [180] | Symmetric filters | 🟢 | global | FP |  | sample |  |  | 40 |
| Liu et al. | [192] | Local symmetries for alignment | 🟡 | global | FP |  | sample |  |  | 14 |
| Liu et al. | [188] | Complex filters | 🟢 | global | FP |  | sample |  |  | 34 |
| Lumini et al. | [196] | PCA on orientation field | 🟢 | global | FP |  | sample |  |  | 116 |
| Maio et al. | [199] | Variation of FingerCode | 🟢 | global | FP |  | sample |  | ✓ | 25 |
| Ross et al. | [246] | 8 Gabor filters of tessellation | 🟢 | global | FP |  | sample |  | ✓ | 465 |
| Turky et al. | [289] | SOMs for representation of OFs | 🟡 | global | FP |  | sample |  |  | 8 |
| Xu et al. | [313] | Sparse representation | 🟢 | global | FP |  | sample |  |  | 3 |
| Yang et al. | [316] | Invariant moments | 🟢 | global | FP |  | sample |  | ✓ | 8 |

Jain et al. proposed so-called *FingerCode* [135]. The idea is to use eight *Gabor* filters for filtering the fingerprint samples. The eight resulting filter responses is sampled at a tessellation grid over the fingerprint sample. The tessellation grid is circular with its center on a detected reference point. By doing so, all local orientations and their signal quality are encoded into a fixed length vector. This fact makes the features in this approach very descriptive as it uses the advantages of the high intra-class variance. FingerCodes are an appropriate example for visually perceivable and understandable representation, as differences between different FingerCodes can easily be identified even by visual inspection. Ross et al. proposed usage of a square tessellation [246]. In addition, no reference point is required. The fingerprint sample is instead aligned using information on the fingerprint minutiae. This results in a more or less simple description of orientation field for the entire fingerprint sample. Yang et. al. used Discrete Wavelet Transform instead of Gabors filters for [316]. Invariant moments around core regions are used to generate features. Maio et al., Leung et al., and Kavati et al. proposed further variations of the MinutiaCode approach [199][176][147].

Liu et al. proposed to use local symmetries for the alignment of the fingerprint sample [191]. The aligned orientation field is then used to generate the index vector. Symmetries again are perceivable by humans, which facilitates understanding of this approach. Li et al. used similar symmetric filters for description of characteristics in the orientation field [180]. Liu et al. proposed to use complex filters on the orientation field to find singularities [188]. Those singularities are used as features. Of course, challenges will arise, if no singularities are present, e.g. in small fingerprints.

Some approaches describe the orientation field more directly. Lumini et al. proposed to use a *Principal Component Analysis* (PCA) on the orientation field [196]. With respect to orientation fields, this approach was the first to evolve from FC to continuous FI. This is a straightforward approach to generate a compact representation from an orientation field. Xu et al. proposed to use the method of *Total Variation* to reconstruct the assumed orientation field of a fingerprint sample [313]. Total variation is a popular method for modelling smooth vector fields like a fingerprint orientation field. It is quite tolerant to disturbances of the vector field. As orientation field estimation is no trivial task, this approach might be quite robust against quality variations between fingerprint samples. This approach therefore makes use of the fact, that fingerprint orientation fields are smooth. Finally a sparse representation is generated from the reconstructed orientation field. Turky et al. proposed to use *Self-organizing maps* (SOM) for description of the orientation field [289]. SOMs can finally also be used to for generation of the index vector. SOMs have the capabilities to deal with disturbed inputs. This approach may therefore be appropriate for fingerprint samples of low quality.

### 14.4.5 Hybrid Approaches

A few approaches have been proposed, which use features from more than one domain. These approaches in general do not propose new features for generation of index vectors. They only combine features proposed by others. Due to the combination of features and the fact that more information is processed, these approaches may achieve higher accuracies. Another advantage is of course, that if a single sub component like the orientation field estimation is improved, the entire FI may be improved. On the other hand, such hybrid approaches may be prone to failures in the single sub component. Such approaches may unintentionally combine not only the advantages but also the drawbacks of their sub components. In some cases each feature is indexed for its own and in some cases the features are used jointly. All found approaches use a global representation of the features.

Even though these approaches do not provide any new features, they are still worth listing here. Lee et al., Jiang et al., Liu et al., and Cappelli proposed to use the orientation field and ridge frequencies (describing distances between neighbouring ridges) as features for FI [175][142][189][44]. Later, Capppelli et al. extended this approach with the use of MCC [45]. Paulino et al. extended this approach even further [233]. They proposed to use a combination of minutia triplet, MCC, the orientation field, fingerprint singularities, and ridge frequencies. Bazen et al. combined orientation fields and minutia [20]. De Boer et al. used orientation fields, FingerCode, and triplets for FI [63]. Pandey et al. proposed to use minutia triplet, MCC, and the orientation field for FI [230]. All features are compared individually. In the end a fusion of all three results is performed. Han et al. proposed a hashing directly on the fingerprint minutiae and enriched the minutiae information with features extracted from the fingerprint image [103].

**Table 14.6:** Approaches making use of more than one characteristic.

| Authors | Ref. | Approach | Level | Repr | Mod | Multi | Index | $\mathcal{O}(\cdot)$ | ⏱ | Cit. |
|---|---|---|---|---|---|---|---|---|---|---|
| Bazen et al. | [20] | OF and minutiae | 🟢 | global | FP | | sample | | ✓ | 2 |
| de Boer et al. | [63] | OF, FingerCode, and triplets | 🟢 | global | FP | | sample | | | 104 |
| Cappelli | [44] | OF and ridge frequencies | 🟢 | global | FP | | sample | ✓ | ✓ | 46 |
| Cappelli et al. | [45] | MCC, OF, and ridge frequencies | 🟢 | global | FP | | sample | | ✓ | 23 |
| Han et al. | [103] | Minutiae and image | 🟡 | global | FP | | sample | | | 17 |
| Jiang et al. | [142] | OF and ridge frequencies | 🟢 | global | FP | | sample | | ✓ | 80 |
| Lee et al. | [175] | OF and ridge frequencies | 🟢 | global | FP | | sample | | | 22 |
| Liu et al. | [189] | OF and ridge frequencies | 🟢 | global | FP | | sample | | | 68 |
| Pandey et al. | [230] | Triplets, MCC, and OF | 🟡 | global | FP | | feature | | | 0 |
| Paulino et al. | [233] | Triplets, MCC, OF, singularities, and ridge frequencies | 🟢 | global | FP | | feature | | ✓ | 22 |

## 14.5 Usage of Data and Metrics

The International Organization for Standardization (ISO) is currently working on the topic of indexing techniques. But by now, there is no standard to evaluate biometric indexing. Thus, there is also no standard test set. The surveyed works evaluated their proposed approaches on a multitude of different datasets. Table 14.8 gives an overview over the evaluations.

There are datasets publicly available for evaluation. The benchmark series *Fingerprint Verification Competition* (FVC) has four editions, which supply the following volumes of datasets: 2000 [200], 2002 [201], 2004 [202], and 2006 [47]. The National Institute of Standards and Technology (NIST) provides a so-called special database *NIST SD4* of rolled fingerprints [307]. There is a variant of this dataset available, which reassembles a natural distribution of pattern types: *NIST SD4 nat*. NIST also provides another dataset of rolled fingerprint: *NIST SD14*. The University of West Virginia provides the multi-modal data set *WVU*. This set also contains fingerprint samples. All datasets have individual characteristics. Table 14.7 provides information on the characteristics of the datasets. Table 14.8 shows which approach was evaluated on which dataset. Some approaches were evaluated on unknown, sequestered or only rarely used datasets. Those datasets are summarized to the category *Others*. Two datasets were used for testing most often: FVC2002 DB1 and the NIST SD4 or rather its natural subset. Because of their frequent usage, both can be seen as some kind of pseudo standard testsets. In general, no reviewed work gave reasons for selection of the evaluated datasets.

**Table 14.7:** Datasets used for evaluation.

| Dataset | Samples | Dimensions | Resolution | Data Origin | Ref |
|---|---|---|---|---|---|
| FVC2000 DB1 | 800 | 300×300 | 500dpi | Optical | [200] |
| FVC2000 DB2 | 800 | 364×254 | 500dpi | Capacitive | |
| FVC2000 DB3 | 800 | 478×448 | 500dpi | Optical | |
| FVC2000 DB4 | 800 | 320×240 | ~500dpi | Synthetic | |
| FVC2002 DB1 | 800 | 388×374 | 500dpi | Optical | [201] |
| FVC2002 DB2 | 800 | 560×296 | 569dpi | Optical | |
| FVC2002 DB3 | 800 | 300×300 | 500dpi | Capacitive | |
| FVC2002 DB4 | 800 | 384×288 | ~500dpi | Synthetic | |
| FVC2004 DB1 | 800 | 640×480 | 500dpi | Optical | [202] |
| FVC2004 DB2 | 800 | 364×328 | 500dpi | Optical | |
| FVC2004 DB3 | 800 | 480×300 | 512dpi | Thermal Sweeping | |
| FVC2004 DB4 | 800 | 384×288 | ~500dpi | Synthetic | |
| FVC2006 DB1 | 1,680 | 96×96 | 250dpi | Capacitive | [47] |
| FVC2006 DB2 | 1,680 | 560×400 | 569dpi | Optical | |
| FVC2006 DB3 | 1,680 | 500×400 | 500dpi | Thermal Sweeping | |
| FVC2006 DB4 | 1,680 | 384×288 | ~500dpi | Synthetic | |
| NIST DB4 | 4,000 | 512×512 | 500dpi | Ink-based | [307] |
| NIST DB4 natural | 2,408 | 512×512 | 500dpi | Ink-based | |
| NIST SD14 | 54,000 | 768×832 | 500dpi | Ink-based | [306] |
| WVU | 7,219 | 292×248 | 500dpi | Optical | [60] |

When dealing with pattern recognition, *generalization* is an important aspect. Generalization indicates, how well a task can be solved by an approach on unknown data. A method's degree of ability to generalize can be assessed typically in two ways. First, a strict splitting of the data into a part used for training and a part

**Table 14.8:** Checkmarks ✓indicate reported results for a dataset. Brackets indicate that only a subset was tested. Some approaches explicitly state a training set or claim separation of training and test data.

| Feature | Reference | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2000 DB4 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2002 DB4 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2004 DB4 | FVC2006 DB1 | FVC2006 DB2 | FVC2006 DB3 | FVC2006 DB4 | NIST DB4 | NIST DB4 natural | NIST SD14 | WVU | Others | Training Set | Separation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Minutiae | [9] |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ |  | (✓) |  |  |  |  |
|  | [10] |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |
|  | [21] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [23] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [25] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  | ✓ |  |  |
|  | [27] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [52] |  | ✓ | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ | ✓ | ✓ |  |  | ✓ | ✓ |
|  | [56] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [79] |  | ✓ | ✓ |  | ✓ |  |  |  | ✓ |  |  |  |  | ✓ |  |  | ✓ | ✓ | (✓) |  |  |  |  |
|  | [85] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [106] |  |  |  |  | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [114] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ | ✓ |  |
|  | [117] |  |  |  |  | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ | ✓ |
|  | [132] |  |  |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  | ✓ | ✓ |
|  | [131] |  |  |  |  | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [138] |  |  |  |  |  |  |  |  | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [150] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |
|  | [156] |  | ✓ | ✓ |  | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  | ✓ |  | (✓) |  |  |  |  |
|  | [155] |  | ✓ | ✓ |  | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  | ✓ |  | ✓ |  |  |  |  |
|  | [161] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |
|  | [166] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [169] |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [170] |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [178] |  |  |  |  | ✓ | ✓ |  |  | ✓ | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  | ✓ | ✓ |
|  | [78] |  |  |  |  | ✓ | ✓ |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  | ✓ |  |  |
|  | [182] |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [183] |  |  |  |  | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [185] |  |  |  |  | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [204] |  |  |  |  | ✓ |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [211] |  |  | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ |  | ✓ |  |  |  |  |
|  | [212] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ | ✓ |  |  |  |  |  |
|  | [213] |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |
|  | [216] | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [241] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [245] |  |  |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |
|  | [296] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
|  | [297] |  |  |  |  | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [302] |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [303] |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  | ✓ | ✓ |
|  | [312] |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [315] |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [333] |  | ✓ |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |
|  | [334] |  | ✓ |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

for testing. Second, testing on multiple different test sets may reveal the ability to generalize with respect to a larger variety of data. Achieving good results on a single dataset may just be a fluke.

Most of the approaches have tunable parameters. But in almost no case a training set to tune the parameters was declared. It is therefore unclear, whether a strict separation of the data was applied. This in turn allows doubts in the generalization ability of most of the approaches. Many approaches have been tested only on one

**Table 14.9:** Checkmarks ✓indicate reported results for a dataset. Brackets indicate that only a subset has been tested. Some approaches explicitly state a training set or claim splitting training and test data.

| Feature | Reference | FVC2000 DB1 | FVC2000 DB2 | FVC2000 DB3 | FVC2000 DB4 | FVC2002 DB1 | FVC2002 DB2 | FVC2002 DB3 | FVC2002 DB4 | FVC2004 DB1 | FVC2004 DB2 | FVC2004 DB3 | FVC2004 DB4 | FVC2006 DB1 | FVC2006 DB2 | FVC2006 DB3 | FVC2006 DB4 | NIST DB4 | NIST DB4 natural | NIST SD14 | WVU | Others | Training Set | Separation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ridges** | [76] | | | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [108] | | ✓ | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [137] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [139] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [160] | | | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [191] | | | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [318] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | ✓ | ✓ |
| | [326] | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | | |
| | [218] | | | | | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | | | | | | |
| | [268] | | ✓ | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [332] | | | | | | | | | | | | | | | | | ✓ | | | | | ✓ | ✓ |
| **Scores** | [100] | | | | | | | | | | | | | | | | | | | | (✓) | | | |
| | [101] | | | | | | | | | | | | | | | | | ✓ | | | (✓) | | | |
| | [102] | | | | | | | | | | | | | | | | | | | | ✓ | | | |
| | [214] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [51] | | ✓ | ✓ | | ✓ | | | | | | | | | | | | ✓ | | ✓ | | | | |
| **OF** | [135] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [147] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [176] | | ✓ | ✓ | | ✓ | | | | | | | | | | | | (✓) | | | | | | |
| | [180] | | | | | | | | | | | | | | | | | | ✓ | | | | | |
| | [192] | | ✓ | | | | | | | | | | | | | | | | | | | | | |
| | [188] | | | | | | | | | | | | | | | | | ✓ | ✓ | | | | | |
| | [196] | | | | | | | | | | | | | | | | | ✓ | | | | | | |
| | [199] | | | | | (✓) | (✓) | (✓) | (✓) | | | | | | | | | | | | | | | |
| | [246] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [289] | | | | | | | | | | | | | | | | | | | | | ✓ | ✓ | ✓ |
| | [313] | | ✓ | | | ✓ | | | | | | | | | | | | | | | | | | |
| | [316] | | | | | ✓ | | | | | | | | | | | | | | | | | ✓ | ↯ |
| **Hybrid** | [20] | | ✓ | | | | ✓ | | | | | | | | | | | | | | | | | |
| | [63] | | ✓ | | | | | | | | | | | | | | | | | | | | | |
| | [44] | | ✓ | ✓ | | ✓ | | | | | | | | | | | | ✓ | ✓ | ✓ | | | ✓ | ✓ |
| | [45] | | ✓ | ✓ | | ✓ | | | | | | | | | | | | ✓ | ✓ | ✓ | | | | |
| | [103] | | | | | | | | | | | | | | | | | | | | | | | |
| | [142] | | | | | | ✓ | ✓ | | | | | | | | | | | ✓ | | | | | |
| | [175] | | | | | | | | | | | | | | | | | | ✓ | | | | | |
| | [189] | | | | | | | | | | | | | | | | | ✓ | | | | | | |
| | [230] | | | | | | | | | | | | | | | | | | | | | ✓ | | |
| | [233] | | | | | | | | | | | | | | | | | | | ✓ | | ✓ | | |

or two data sets. A larger variety of data would have been desirable. Unfortunately, only very few approaches have been tested thoroughly on a larger number of datasets.

FI is meant to be applied to large datasets. Unfortunately, no really large datasets are publicly available. Some approaches have been tested on large but sequestered datasets. This does not allow reproducibility of the claimed results.

Besides the aspect of datasets, metrics are also important when assessing an approach. There are several policies on how to generate the candidate list (see section 14.2). Each policy has its own reasonable metrics. In general, no approach was evaluated with respect to all policies. Usually, only a single policy was evaluated.

This results in a multitude of used metrics, which are not comparable directly. In addition, even the naming of the metrics is not following a standard. This results in the confusing usage of a multitude of synonyms for metrics. Two metrics and their synonyms are used most frequent: *Penetration rate* and *Error rate*. The Penetration rate is the ratio between the length of the candidate list and the entire database. The Error rate is the ratio of candidate lists, which do not contain the genuine candidate. Both rates usually depend on ranks in an ordered candidate list.

It is worth mentioning, that there is an independent benchmark for FI: FVC-ongoing [68]. This benchmark evaluates on a large, sequestered dataset. In addition, it would provide common metrics for evaluation. Thus, FVC-ongoing would allow reasonable comparison of approaches. Unfortunately, there is only one approach with published results.

## 14.6 Conclusion

FI can be a key processing step when dealing with large fingerprint databases. Various approaches for FI were proposed in the past. This work surveyed the approaches, which were found in the four relevant archives. The approaches can be grouped into five categories with respect to the features which are processed. Most approaches use fingerprint minutiae as input features. A few approaches work on ridges/textures, orientation fields, or biometric scores. There are also some hybrid approaches.

It is almost impossible to identify a state-of-the-art in FI for several reasons: First, there is no standard protocol or common metric. Usually Error Rate is evaluated against Penetration Rate. Second, there is no standard dataset for evaluation. Even though, the datasets FVC2002 DB1 and the NIST SD4 are the most commonly used, only about half the approaches evaluated on these datasets. Some approaches even evaluated on sequestered data. Claimed results are therefore not reproducible at all. Only very few approaches were evaluated on large datasets even though this would be the use case for FI. Third, there is a lack of independent external evaluation, even though the benchmark FVC-ongoing would be available for this very task.

The level of description of the proposed approaches is quite good in most of the reviewed works. This allows a fair chance of reimplementation of the approaches.

Unfortunately, several deficiency in the quality of the body of research have been observed. In almost all approaches the methods have tunable parameter. But only in very few cases, the dataset used for optimization of those parameters is declared. This lack of declaration makes it impossible for those approaches to claim separation of training and test data. Having no separation between training and test data allows no conclusion on generalization of the approaches. But generalization is an important aspect in pattern recognition.

The aspect of computational complexity is a central aspect of FI. Even approaches allowing lowest penetration rates are worthless, if computations just take too much time. A very large fraction of found publications simply neglects this aspect. After all, about half of the approaches report some kind of timing of their approaches - even though this is only a weak proxy for the computational complexity.

Bit sizes of index vectors are especially of interest when databases are large, which is essentially the use case for application of FI. Almost no approach explicitly reports this aspect. However, thanks to quite good descriptions of the approaches, bit sizes are derivable.

# Chapter 15

# Unsupervised Learning of Fingerprint Rotations

**Summary** Alignment of fingerprint samples can be a useful preprocessing steps in fingerprint recognition. Designing approaches for alignment usually requires knowledge on ground truth alignments. An unsupervised training strategy for CNNs is presented, which does not depend on unreliable ground truth. A proof of concept is demonstrated on several datasets.
This publication is joint work with Jan Marek May and Christoph Busch. It was presented at the *BIOSIG* in Darmstadt (Germany) in 2018.

[264] P. Schuch, J. M. May, and C. Busch. Unsupervised learning of fingerprint rotations. In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553096

**Abstract** The alignment of fingerprint samples is a preprocessing step in fingerprint recognition. It allows an improved biometric feature extraction and a more accurate biometric comparison. We propose to use Convolutional Neural Networks for estimation of the rotational part. The main contribution is an unsupervised training strategy similar to Siamese Networks for estimation of rotations. The approach does not need any labelled data for training. It is trained to estimate orientation differences for pairs of samples. Our approach achieves an alignment accuracy with a mean absolute deviation 2.1° on data similar to the training data, which supports the alignment task. For other datasets accuracies down to 6.2° mean absolute deviation are achieved.

## 15.1 Introduction

Despite the fact that fingerprint recognition is a mature and widely deployed technology, it still can be improved or extended to uprising demands. Some aspects in fingerprint recognition can benefit from *aligning* fingerprint samples to a common orientation and positioning. Both concerns can be tackled separately [203]. This work focusses on the orientation part of the alignment. There are two categories of alignments: alignments common for all fingerprints and those shared for all samples of the same finger. The former is the more general approach. The latter allows a higher degree of freedom in the alignment, since each fingerprint may have its individual alignment. Such an alignment can be beneficial in many cases. Most important, biometric comparison algorithms may benefit and especially speed up since they do not have to deal with large rotations between compared samples.

A trivial orientation alignment is the upright position in the direction of the fingertip. But finding this trivial orientation is actually far from being trivial. Usually, fingerprint alignment works on *focal points* [243]. Detection of such points is challenging, if the fingerprint quality is low or there is only a partial print without any focal points in it. Another serious challenge in fingerprint alignment is the fact there is no *ground truth* for the alignment, i.e. you never know for sure what the correct alignment is. No assumptions on *initial rotations* of fingerprint samples can be made. Any manual labelling is prone to inaccuracies and lacks reproducibility.

Deep learning has provided quite impressive solutions in many domains of digital image processing and pattern recognition in the last years. We propose to use *Convolutional Neural Networks* (CNN) for the task of fingerprint alignment. Our main contribution is a training strategy. Similar as in *Siamese Networks* the CNN learns a rotational distance between two arbitrarily rotated instances of a single fingerprint sample. As the approach is not dependent on any ground truth data, the training can be done unsupervised. Large amounts of unlabelled data in turn allow to train the CNN. The training finally yields an orientation assignment, which is individual to each finger.

The rest of this paper is organized as follows: Section 15.2 provides an overview on related work. Our approach will be explained in Section 15.3. Section 15.4 explains the experiments, which were carried out to test the proposed approach. Section 15.5 concludes the findings. An outlook on future work can be found in Section 15.6.

## 15.2 Related work

Several approaches for the task of fingerprint pose estimation have been proposed in the past. Sood and Kaur provide an overview on alignment methods in fuzzy vault [271].

Markert et al. proposed to use the almost parallel ridges above creases for alignment, as those ridges usually all have similar directions in every fingerprint [205].

Most of the approaches make use of the orientation field of a fingerprint. The orientation field is a representation for the local orientations of the ridges in a fingerprint. If no original fingerprint sample is available, the orientation field can also be estimated for extracted fingerprint minutiae [162].

Some approaches work directly on the orientation fields. Yang et al. proposed to learn dictionaries of orientation field patches [319]. They used the orientations fields to perform a pose estimation for the fingerprint. Hotz proposed to extract an intrinsic coordinate system based on a longitudinal axis [111]. The longitudinal axis can be found by searching for symmetries in the orientation field. This axis could also be used for a rotational alignment.

Other approaches extract distinctive points from the orientation field. So-called *singularities* (cores and deltas) can be used for an alignment [135]. For fingerprints lacking singularities, *focal points* can be defined as those points with the highest curvature [243]. Nagar et al. and Zhang et al. also used points of maximum curvature as reference points for an alignment [215][330]. Li et al. proposed to use isosceles triangles for alignment [177]. Isosceles triangles are placed on the ridges. The approach makes use of local symmetries near focal points. Liu et al. proposed a multi-scale approach for detection of the focal points from orientation fields [187]. Tams proposed to extract reference points from the fingerprint orientation field [281]. Jain and Minut used kernel curves to describe the flow of the fingerprint ridges [134]. Those kernel curves describe the behaviour of the ridges

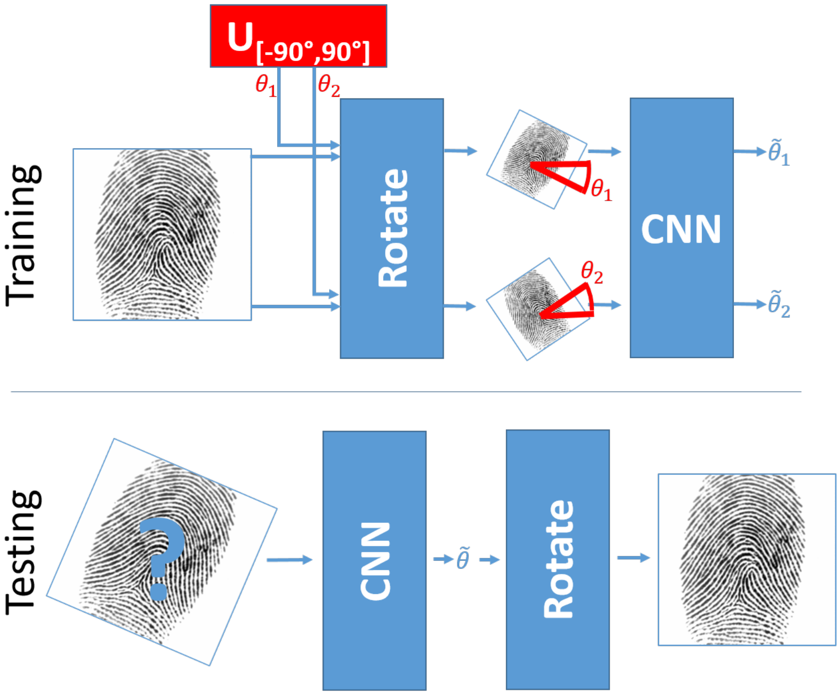


**Figure 15.1:** During training a fingerprint sample is rotated by two random angles $\theta_1$ and $\theta_2$. The CNN estimates rotations $\tilde{\theta}_1$ and $\tilde{\theta}_2$, so that differences ($\theta_1$-$\theta_2$) and ($\tilde{\theta}_1$-$\tilde{\theta}_2$) are as similar as possible. During testing estimation $\tilde{\theta}$ can be used for rotational alignment.

around the focal points. Best fitting kernel curves can be used for an alignment to typical patterns. Li et al. proposed to estimate an alignment by topological structures around cores [181]. Detection of singularities and focal points depends on an accurate extraction of the orientation field. Such a detection may fail for partial fingerprint samples.

Merkle et al. proposed to use the shape of the fingerprint sample [207]. The outline of the fingerprint sample usually can be approximated by an ellipse. This ellipse is aligned to its principal axis. Yang et al. proposed to align small neighbourhoods of minutiae [314].

There is also an approach, which uses techniques from the domain of Deep Learning (DL). Ouyang et al. proposed to use a variation of a *Region-based Convolutional Network* (R-CNN), which have originally been proposed to the task of object detection [229]. The authors compared the estimated poses to manually labelled ground truth poses. They have also found positive effects of alignment when applying *Fingerprint Indexing*.

There are further approaches in Deep Learning for the task of general alignment, e.g. *Spatial Transformer Networks* proposed by Jaderberg et al. [130]. Those can learn transformations of sampling grids for given tasks. The original image is then sampled at the transformed sampling grid. Any transformation, which helps solving the given task, can be learned in this approach. Our approach does not need a given task for training.

## 15.3 Our Approach

### 15.3.1 CNNs for Estimations of Rotations

We will exploit a training strategy called *Siamese Network* [34] for the task of learning fingerprint alignment. Each training input consists of a pair of randomly rotated instances of a single fingerprint sample (see Figure 15.1). The CNN estimates a rotation for each instance. The CNN shall learn to assign estimations in such a way, that the difference between the two estimated values is close to the difference of the two actual rotations. The main advantage of this approach is, that no expert based ground truth rotational information is necessary at all. This allows to use any unlabelled fingerprint sample for training. For testing the trained CNN estimates the rotation for a given input fingerprint sample (see Figure 15.1). This estimation can be used to align the input fingerprint sample by a corresponding rotation.

### 15.3.2 Architecture

There was no extensive optimization of any hyperparameters. The entire model assembles seven different types of layers: *Convolutional* layers (ConvLayer), *Parametric Rectified Linear Units* (PReLu), *Batch Normalization* layers (BatchNorm), *Maximum Pooling* layers (MaxPooling), *Flatten* layers (Flatten), *Dense* layers (Dense), and *Softmax* layers (Softmax). There was no *striding* in the ConvLayers. Table 15.1 gives an overview over the entire architecture and the outputs of each layer.[1] The CNN has $179,424$ trainable parameters. The input is a grey scale image. A region of interest of size $192 \times 192$ pixels is cropped from the image's center.[2] This size allowed reasonable performance for images of about $500\,\text{dpi}$ resolution. It also allows application to small images, e.g. $300 \times 300$ images in FVC2000DB1 [200].

Rotation estimation is obviously a regression task. However, it was shown that classification can be superior over regression even on regressions task, e.g. if one

[1] The model was created and trained in the DL framework *Tensorflow* [1].

[2] Cropping the region of interest to the center of the fingerprint sample's foreground might be beneficial. However, this approach is independent from such a foreground detection.

| # | Layer | Output |
|---|---|---|
| 0 | Input (192×192×1) | (192×192×1) |
| 1 | ConvLayer (32×5×5×1) | (192×192×32) |
| 2 | BatchNorm | (192×192×32) |
| 3 | PReLU | (192×192×32) |
| 4 | MaxPooling (4×4) | (48×48×32) |
| 5 | ConvLayer (32×5×5×32) | (48×48×32) |
| 6 | BatchNorm | (48×48×32) |
| 7 | PReLU | (48×48×32) |
| 8 | MaxPooling (4×4) | (12×12×32) |
| 9 | ConvLayer (32×5×5×32) | (12×12×32) |
| 10 | BatchNorm | (12×12×32) |
| 11 | PReLU | (12×12×32) |
| 12 | MaxPooling (4×4) | (3×3×32) |
| 13 | Flatten | (288) |
| 14 | Dense (256×288) | (256) |
| 15 | PReLU | (256) |
| 16 | Dense (256×256) | (256) |
| 17 | PReLU | (256) |
| 18 | Dense (181×256) | (181) |
| 19 | Softmax | (181) |

**Table 15.1:** CNN's architecture. The final output can be interpreted as probabilities for different alignment angles.

uses *Deep Expectation*, which averages estimations over classes to form a better estimation [247]. The method of Deep Expectation has also already been successfully applied to the domain of fingerprint recognition: It was used for fingerprint orientation field estimation [260]. Each *class* in the classification approach represents a distinctive angular offset. The outputs of the Softmax layer in this CNN can be interpreted as probabilities for the angular offsets. Weighted averaging according to the probabilities yields the final estimation. We tried regression and classification for the task of estimating a rotational alignment. Our experiments yielded classification including Deep Expectation to be more effective than regression for this task.

### 15.3.3 Training

Training of the CNN can be roughly summarized as follows: The CNN's task is to learn the rotational difference between two randomly rotated instances of the same fingerprint (see Figure 15.1).

In detail, during training a fingerprint sample is picked from the training data. Two random values $\theta_1$ and $\theta_2$ for rotations are sampled from a uniform distribution $\mathcal{U}_{[-90°,90°]}$. The fingerprint sample is rotated according to $\theta_1$ and $\theta_2$. Those two rotated instances are then fed into the CNN, which calculates two estimations $\tilde{\theta}_1$ and $\tilde{\theta}_2$ respectively. We cannot make any assumptions about the *initial rotation* of the original fingerprint sample. However, $\theta_1$ and $\theta_2$ must be assumed to be biased by this initial rotation. Therefore, minimization of the differences $|\theta_1 - \tilde{\theta}_1|$ and $|\theta_2 - \tilde{\theta}_2|$ likewise does not seem reasonable. To circumvent the lack of knowledge on the initial rotation, we formulate the loss function using the difference between the random rotations and the difference between the estimations:

$$\text{loss}\left(\theta_1, \theta_2, \tilde{\theta_1}, \tilde{\theta_2}\right) = \left|(\theta_1 - \theta_2) - (\tilde{\theta_1} - \tilde{\theta_2})\right| \tag{15.1}$$

By doing so, the CNN learns the rotational difference between two rotations. Minimization $|\theta_1 - \tilde{\theta}_1|$ and $|\theta_2 - \tilde{\theta}_2|$ would have a single optimal solution $\{(\theta_1^*, \theta_2^*) : \theta_1^* = \theta_1, \theta_2^* = \theta_2\}$. The proposed loss function allows an infinite set of optimal solutions $\{(\theta_1^*, \theta_2^*) : (\theta_1^* - \theta_1 = \theta_2^* - \theta_2)\}$. A training sample therefore consists of two random rotation angles and the corresponding rotated instances of the same fingerprint sample. By the way, using two different samples would fail, since their initial rotation is unknown.

Eight pairs of samples were processed as a single batch. We used optimizer *AdaGrad* [70]. Other optimizers like *Stochastic Gradient Descent* or *Adam* did not perform significantly different. Learning rate was set to $10^{-5}$.

Training was stopped, when the mean absolute deviations of a validation set did not improve any further. This was done to prevent over-fitting to the training data. Training took only a few hours on a GPU[3]. Estimation of the rotation for a single fingerprint sample takes about 4ms on GPU.

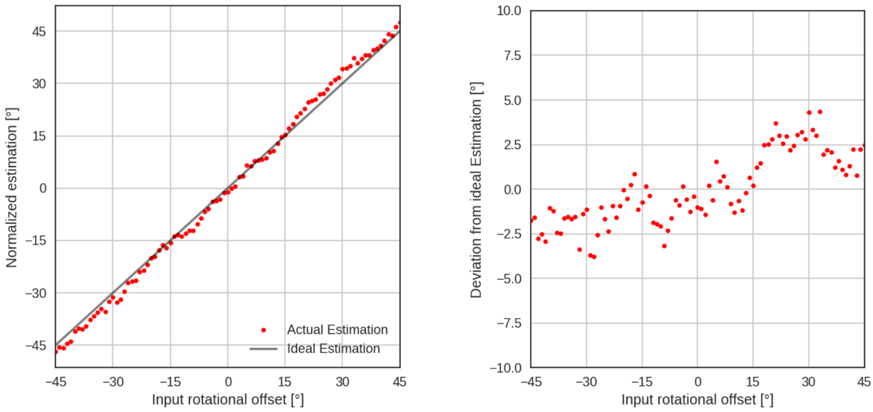
### 15.3.4 Data Augmentation

For CNNs holds in general, that the more appropriate training data is available, the higher is the performance. A common method to increase the amount of training data is *data augmentation*. Rotating the input data according to the random angular distortions already increases the amount of available training data. In addition, we did some additional augmentation by slightly shifting the input image horizontally and vertically. This is a way to prevent *over-fitting*, i.e. the CNN works significantly better on the training data than on any other data, which has not been seen during training.

## 15.4 Experiments

We tested our approach by training the CNN on fingerprint samples from dataset FVC2002 DB1[201]. The dataset consists of 100 fingers with eight impressions

[3] NVIDIA GTX 780



**(a)** **(b)**

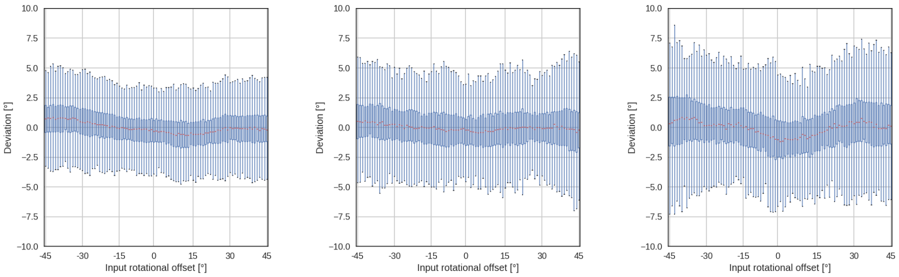
**Figure 15.2:** For testing, the CNN estimates rotations for rotated instances of a given fingerprint sample. Normalization allows to ignore the unknown initial offset of a single sample (15.2(a)). A linear regression with unit slope represents an ideal estimation. It allows inspection of the actual deviations (15.2(b)).

each. We split the set of 800 fingerprint samples into three parts: the first 400 samples for training, the next 200 samples for validation, and the last 200 samples for testing. By doing so, we took care, that no finger is shared between the sets.

No assumptions on distributions of rotations in the real world can be made. We therefore set up a test scenario, in which each fingerprint sample was rotated by angels from $[-45°, 45°]$. Quantisation step for the angles was $1°$. The trained CNN was then applied to the rotated fingerprints to estimate the rotational offsets. Only differences between rotations are relevant. Plain estimations cannot be used for evaluation. A normalization is necessary. Let $\mu_\theta$ be the mean of the estimations for all rotated samples of the same fingerprint. We can then calculate a *normalized* estimation $\tilde{\theta}_N(\alpha)$ for estimation $\tilde{\theta}(\alpha)$ for a sample rotated by $\alpha$ by subtracting the mean estimation $\mu_\theta$:

$$\tilde{\theta}_N(\alpha) = \tilde{\theta}(\alpha) - \mu_\theta \tag{15.2}$$

The normalized estimation $\tilde{\theta}_N(\alpha)$ is independent from any initial rotations in the original fingerprint sample. Thus, the normalization yields evaluations in the first place. Then we calculated a linear regression with a unit slope for the normalized estimations for all rotated instances of the fingerprint (see Figure 15.2). The linear regression represents an ideal estimation. Therefore, the linear regression can be



**(a)** **(b)** **(c)**

**Figure 15.3:** Visualization of deviations between input orientations and normalized estimation for the entire datasets(see Figure 15.2 for deviations of a single sample).

used to evaluate the estimation independent from the unknown initial rotation of the original fingerprint sample.

Linear regressions are calculated for each fingerprint sample in a dataset. The box plot in Figure 15.3(a) visualizes the deviations between estimations and ideal estimations for all samples in the training set. The deviation of the mean for the entire dataset varies about $1^\circ$ around the ideal estimation. The relative deviation is therefore small compared to the range of the input rotations. The normalized deviation $\tilde{\theta}_N(\text{inp}_\alpha)$ of a given input sample *inp* rotated by angle $\alpha$ can be used to calculate the *mean absolute deviation* $\delta$ over M samplings:

$$\delta(\text{inp}) = \frac{1}{M} \sum_{\alpha=-45^\circ}^{45^\circ} \left| \alpha - \tilde{\theta}_N(\text{inp}_\alpha) \right| \tag{15.3}$$

For a given dataset with N input samples *inp* and a given range of input rotations we can calculate the mean deviation $\bar{\delta}$ per dataset:

$$\bar{\delta} = \frac{1}{N} \sum_{i=1}^{N} \delta(\text{inp}_i) \tag{15.4}$$

This value $\bar{\delta}$ can be used to estimate the expected deviation on a dataset, i.e. how well the estimation works on a dataset. Using the proposed evaluation method, we evaluated the trained CNN on the training part, the validation part, and the test part of dataset FVC2002DB1. We extended our evaluations also to other datasets to test the CNN for its generalisation capabilities. We therefore chose beyond FVC2002DB1 five additional datasets from the *Fingerprint Verification Contest* (FVC) benchmark series, which also consists of fingerprint samples from optical

| **Data** | **Trained on**<br>**FVC2002 DB1**<br>**Mean Abs Dev $\bar{\delta}$** | **Trained on**<br>**FVC2000 DB3**<br>**Mean Abs Dev $\bar{\delta}$** |
|---|---|---|
| FVC2002 DB1 Training | 1.3° | - |
| FVC2002 DB1 Validation | 1.8° | - |
| FVC2002 DB1 Test | 2.1° | 17.2° |
| FVC2000 DB1 | 17.2° | 35.0° |
| FVC2000 DB3 Training | - | 1.6° |
| FVC2000 DB3 Validation | - | 2.0° |
| FVC2000 DB3 Test | 20.8° | 2.6° |
| FVC2002 DB2 | 13.9° | 40.4° |
| FVC2004 DB1 | 6.2° | 13.2° |
| FVC2004 DB2 | 13.4° | 15.4° |

**Table 15.2:** Results when trained on FVC2002 DB1 and FVC2000 DB3.

fingerprint livescanners with a resolution of approx. 500 dpi: FVC2000 DB1 and DB3 [200], FVC2002 DB2 [201], and FVC2004 DB1 and DB2[202]. Only the last 200 fingerprint samples from each dataset were taken for evaluation.

Table 15.2 summarizes the results, when the CNN is trained on the training set of dataset FVC2002 DB1. Keeping in mind, that the quantisation step for estimation classes was 1°, the trained CNN achieves a very small mean absolute deviation of only one degree on the training data. While the approach achieves a mean absolute deviation $\bar{\delta}$ of 1.8° on the validation data, 2.1° are achieved on the test data. Figure 15.4 visualizes eight samples of a single fingerprint. The trained CNN estimated the rotation in the original samples and the samples were rotated according to the estimations.

When applied to other datasets the empirical mean absolute deviations $\bar{\delta}$ range from 6.2° for dataset FVC2004 DB1 to 20.8° for dataset FVC2000 DB3. Fingerprint samples in these datasets have other characteristics than the samples in the training data. A distribution shift in the input data is a typical challenge in pattern recognition. Figure 15.5 visualizes the cumulative probabilities for absolute deviations between the normalized estimations and the ideal rotations.

As the CNN performed worst on FVC2000 DB3, we wondered whether our approach is applicable to this data at all. If training is successful, the distribution shift can be tackled by training on the relevant data. We therefore trained another CNN for estimations on this database. It achieved a mean absolute deviation of 2.6° on the test part of FCV2000DB3 (see Table 15.2). It can therefore also be applied to this dataset.

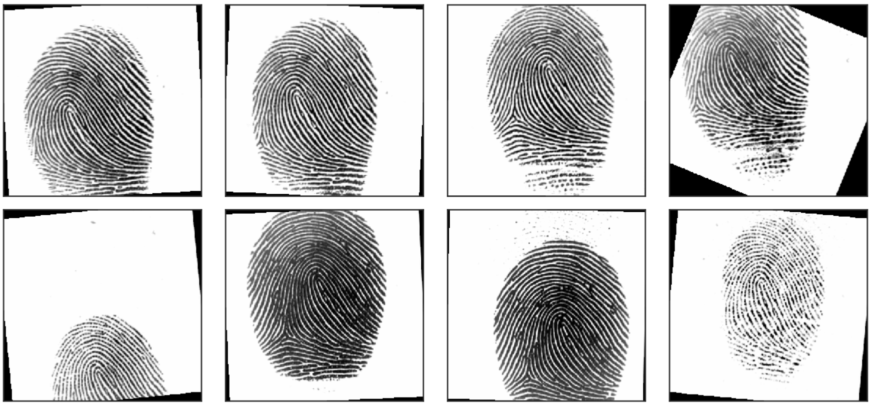


**Figure 15.4:** Eight samples of a single fingerprint from the test set were rotated according to corresponding estimations by the trained CNN.

| Ref | Approach | Dataset | Performance |
|---|---|---|---|
| [111] | Longitudinal axis | NIST SD4 | $4.2°$ mean difference |
| [177] | Isosceles triangles | CASIA | $4.1°$ mean difference |
| [207] | Ellipse from outline | sequestered | $2.3°$ mean difference |
| [229] | R-CNN | NIST SD14 | $95\%$ samples with difference $< 5°$ |
| [319] | Localized dictionaries | NIST SD27 | $13.8°$ mean difference |

**Table 15.3:** Results reported in related work.

Comparisons to other approaches is hardly possible, since there are no other unsupervised approaches for estimation of rotations. Unfortunately, most of the approaches in the related work treat the aspect of rotational alignment as part of a larger workflow and do not explicitly report results for the estimation of the rotation. All reported results from related work can be found in Table 15.3. Comparison is still difficult, since different metrics and different datasets are used for evaluation. However, the reported results allow some comparison to our proposed approach. Hotz et al. achieved a mean difference of $4.2°$ on dataset NIST SD4[307]. Li et al. tested their approach on 80 samples with pattern type *Arch* from dataset CASIA. They reported a mean difference of $4.1°$. Merkle et al. achieved a mean difference of $2.3°$ on a sequestered dataset. Ouyang et al. reported a deviation smaller than $5°$ for $95\%$ of the tested samples of dataset NIST SD14[308]. Yang et al. achieved a mean difference of $13.8°$ on the latent fingerprint dataset NIST SD27[82].

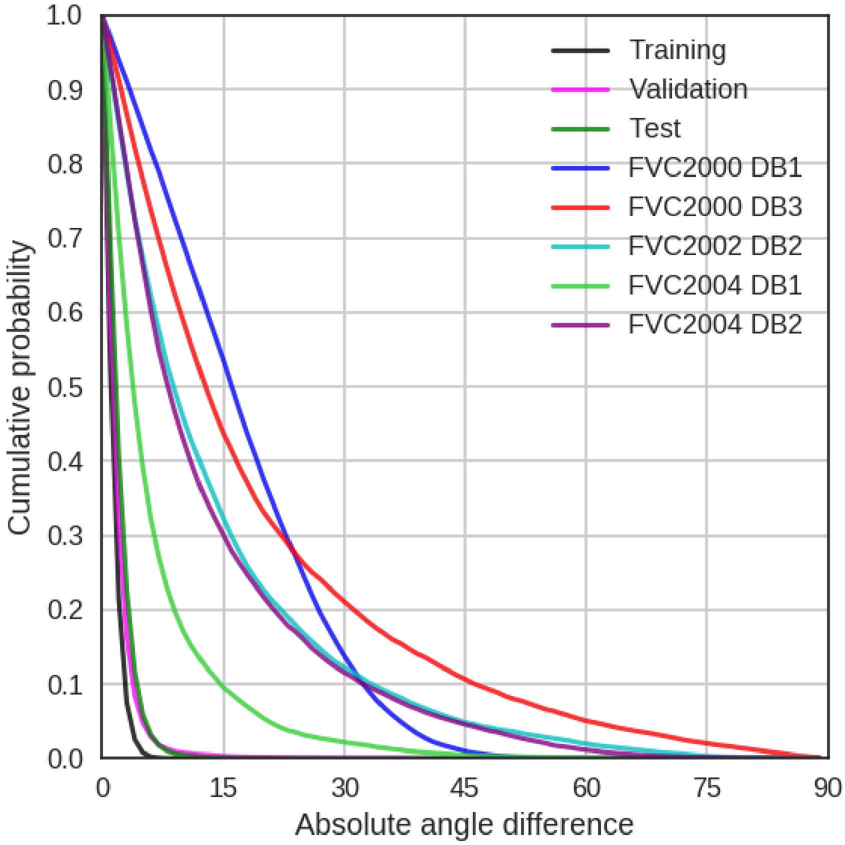


**Figure 15.5:** Cumulative probabilities of exceeding absolute angle differences when trained on dataset FVC2002 DB1.

## 15.5 Conclusion

We presented an approach to estimate a rotational alignment for fingerprint samples. The fingerprints are not guaranteed to be upright after alignment. CNNs were trained to estimate an individual rotational offset for each fingerprint. Our main contribution is the definition of a dedicated loss function, which yields an unsupervised training, independent of any initial rotation of the samples. Application of this loss function is not limited to fingerprint samples. It can be applied to any image data.

The proposed approach was tested on several datasets containing plain fingerprint samples acquired with livescanners. For data similar to the training data a relative small mean absolute deviation of 2.1° to an ideal alignment can be achieved. The best mean absolute deviation for other datasets is 6.2°. Generalization therefore is an issue. However, if enough appropriate training data is supplied, then the approach can be applied to new data.

## 15.6 Future Work

Our approach does only assign an individual rotation to every finger. Extending this approach to train for an absolute rotation alignment may be even more beneficial. In addition, this approach does also tackle only the orientation part of the alignment task. Extending this approach with a translational alignment and scaling would beneficial as well.

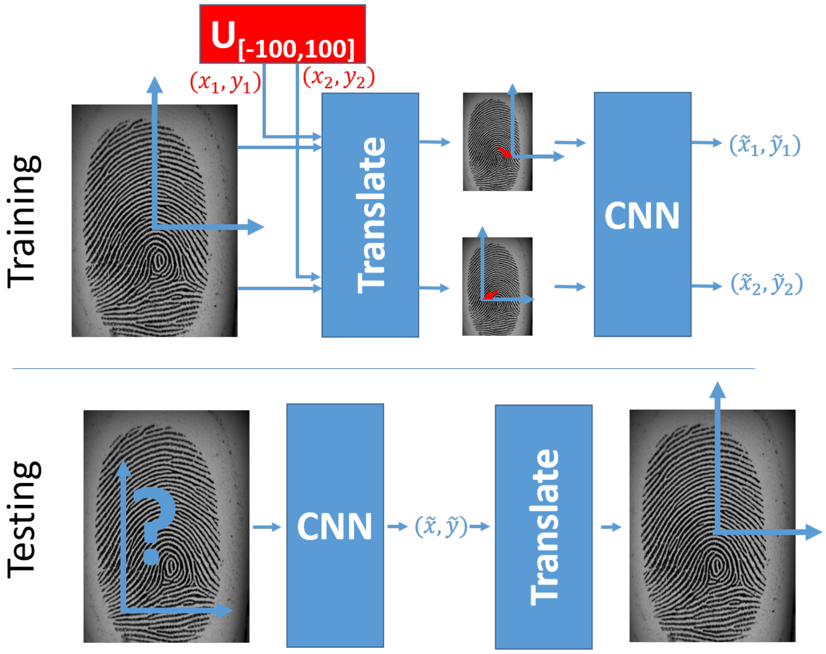


**Figure 15.6:** Similar strategies may allow to learn translational offsets.

A similar unsupervised learning strategy for the translational offsets or the scaling shall be applicable (see Figures 15.6 and 15.7). An obvious strategy to further improve the orientation estimation is to apply the CNN to several crops of a single fingerprint sample and apply a voting strategy among the resulting estimations.

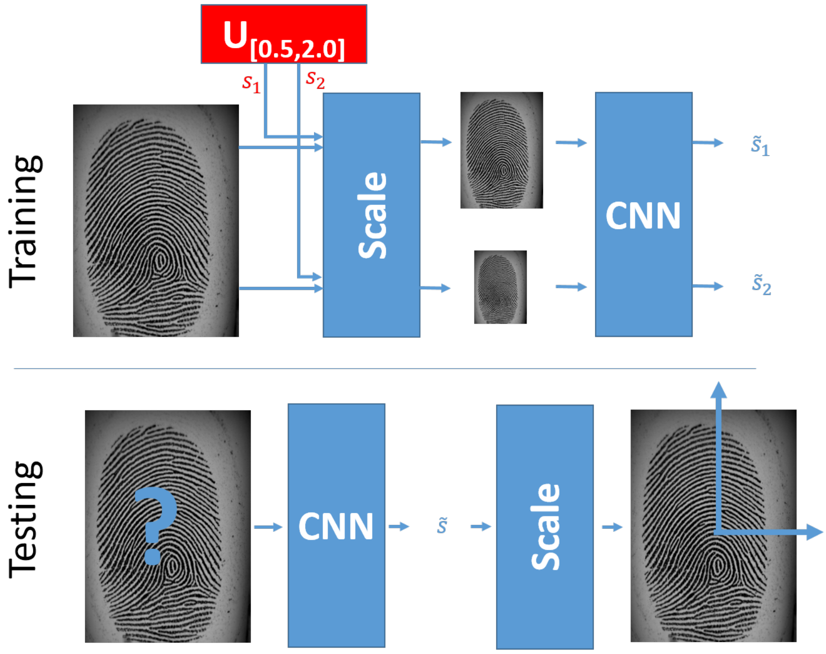


**Figure 15.7:** Similar strategies may allow to learn scaling.

# Chapter 16

# Learning Neighbourhoods for Fingerprint Indexing

**Summary** Fingerprint Indexing can reduce workload of a fingerprint recognition system by generation of candidate lists. Our approach DIMI allows to learn index vectors from ISO/IEC minutiae. Index vectors are only 16 float values per sample, which allows fast comparison. DIMI was tested at FVC-ongoing and achieved the best average candidate list length in the incremental search scenario.
This publication is joint work with Jan Marek May and Christoph Busch. It was presented at the *Signal Information Technology & Internet Based Sytems* conference (SITIS) in Las Palmas de Gran Canaria (Spain) in 2018.

[263] P. Schuch, J. M. May, and C. Busch. Learning neighbourhoods for fingerprint indexing. In *2018 14th Signal Image Technology and Internet-based Systems (SITIS)*, pages 1–6, Nov 2018

**Abstract** Fingerprint Indexing allows filtering databases for samples most similar to a query fingerprint. Fixed-length feature vectors invariant to the count of minutiae allow fast comparison. This is the first approach, which uses Deep Convolutional Neural Networks to generate fixed-length index vectors from minutia neighbourhoods. It is also invariant to any sorting of minutiae with respect to their vicinity. No fingerprint image data is required. Usage of only standardized ISO/IEC minutia templates allows even to deploy to legacy systems. Our approach is therefore called *Deep ISO Minutiae Indexing* (DIMI). Index vectors can be as short as 16 float values per sample. DIMI was evaluated at the independent benchmark framework FVC-ongoing. In the incremental search scenario, DIMI achieves an average penetration ratio of 0.747% for test FIDX-50K and 0.723% for test FIDX-10K respectively. These are the best results reported so far.

## 16.1 Introduction and Motivation

Running fingerprint recognition systems demands specific requirements on the response time. Thorough biometric comparison of two fingerprint samples is a tedious job and can be very time consuming. If it takes longer to identify an individual than acceptable for a given scenario, the system is simply unusable. The idea of *Fingerprint Indexing* (FI) is to sort the fingerprint database beforehand [203]. While some fingerprints are quite similar, some are obviously different. In many cases one can decide at first glance, whether some fingerprints are not mated, e.g. if pattern types between two fingerprints differ. FI excludes those from thorough comparison.

The sorting of the database is achieved by a new and compact representation of the fingerprints. Every item in the database is represented by an index vector, which can be interpreted as a vague description of the fingerprint. FI compares index vectors of query fingerprints with those in a database. This comparison should be a computationally cheap operation compared to the thorough comparison. The output of the index comparison is a candidate list $\mathcal{C}$. This list contains only fingerprints similar to the query fingerprint.

Fingerprint minutiae are the most common features used for fingerprint recognition. Minutiae are salient points in the ridge structure of the fingerprint, e.g. endings and bifurcations of ridges. The location and topology of those minutiae are assumed to be unique for every fingerprint. Unfortunately, the count of minutiae found in every fingerprint sample might be different. Finding a fixed-length representation for an input of variable length is not trivial.

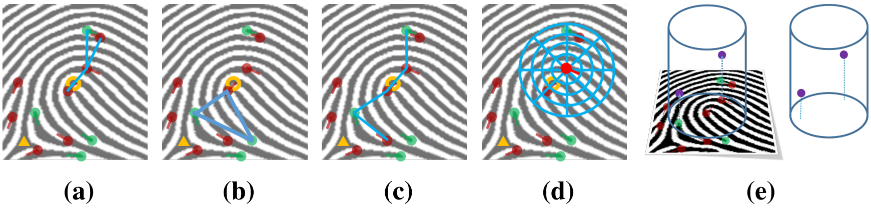


**Figure 16.1:** Minutiae (endings ●- and bifurcations ●-) and singularities (deltas △ and cores ○) can be used for FI in various representations: e.g. nearest neighbours (16.1(a)), triplets (16.1(b)) or sequences (16.1(c)). In some cases the neighbourhood is sampled at a tessellation grid (16.1(d)). *Minutia Cylinder Code* transforms each neighbourhood into a single index vector (16.1(e))

Convolutional Neural Networks (CNN) can be trained to find an appropriate representation for many tasks. CNNs have shown superiority over hand-crafted representations and are state of the art in many domains of image processing and pattern recognition, e.g. the benchmark *ImageNet*[249]. CNN architectures like *PointNet* enable to learn fixed length representations from the variable count of inputs [236]. We adapt and extend it to work on minutiae neighbourhoods. By using *maximum pooling* at different stages of the CNN, representations for neighbourhoods are learned. Those are pooled to form representations for entire fingerprint samples. Our approach is therefore called *Deep ISO Minutiae Indexing* (DIMI), since we are using only ISO/IEC compliant minutia templates[121] as input to the CNN. Thus, our approach can be applied to deployed biometric enrolment databases, in which no image data is available.

Numerous FI approaches were proposed in the past. This is the first one to learn a fixed-length representations from minutiae using CNNs. DIMI was successfully evaluated at independent benchmark framework FVC-ongoing[68].

The rest of this work is organized as follows: Section 16.2 gives an overview on related work. Our approach is described in Section 16.3. Section 16.4 describes the results achieved with our approach on test data. A conclusion can be found in Section 16.5.

## 16.2 Related Work

Various approaches using minutiae were proposed for FI. Figure 16.1 outlines approaches to FI. Approaches can be divided into local and global approaches.

Most approaches work in a local manner. Those approaches are working on sets of few minutiae. Features are designed, which describe the relation of the minutiae within such set. Triplets of minutiae can be represented by their geometric features, e.g. angles and side lengths (see Figure 16.1(b)) [85]. There are many variations and extensions of this approach[161][25][241][333][245][27][155]. Delaunay triangulation is a common way to select triplets [21][182][183][296][212][213][211][156][150][79][179]. Some approaches use more than three minutiae, e.g. quadruplets [297] [114]. There are also approaches combining triplets and quadruplets [117].

Cappelli et al. proposed to represent minutiae and their relative neighbourhood in a compact form called *Minutia Cylinder Code* (MCC) [49]. In this approach a cylinder describes the relative position and relative angle of the neighbouring minutiae (see Figure 16.1(e)). MCC can be considered as state of the art for FI. It is the only approach, which is evaluated at benchmark FVC-ongoing. In addition, there are numerous variations of MCC [303][10][334][11]. *Minutia Disks* describe neighbourhoods in polar coordinates [178][78].

Further approaches describe the local neighbourhoods of minutiae (see Figure 16.1(a)) [166][204][9][23][138][56]. Those approaches usually use the nearest neighbouring minutiae around a *central* or an *anchor minutia*. Series of neighbouring minutiae concatenated to sequences can also be used to form a description (see Figure 16.1(c)) [106].

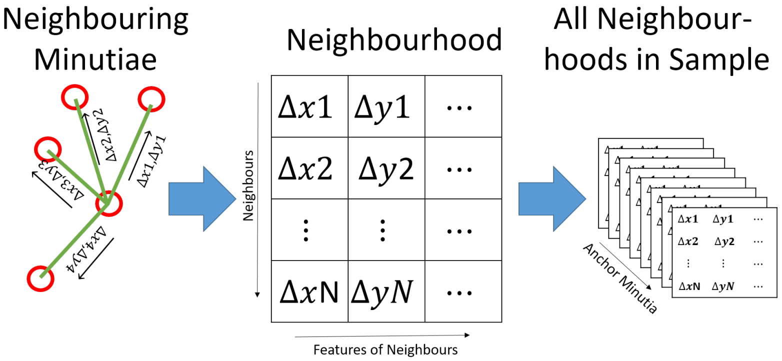
Some approaches focus more on the index generation than on the features [169][170] or the index comparison [302] [216]. Other approaches use minutiae to generate a global representation. This can be achieved by geometric representations [132][131], or a complex representation [312], or random look-up tables [315].

*PointNet* is an approach to apply CNNs to point clouds with a variable number of points [236]. The key here is to apply maximum pooling. This yields invariance to the number of points. *PointNet++* extends this to pairs of points [237]. We will extend the approach of PointNet to deal with neighbourhoods of fingerprint minutiae. Our approach is the first to use CNNs for generation of fixed length index vectors from fingerprint minutiae. It is invariant to the number of input minutiae and invariant to the order of minutiae in each neighbourhood.

## 16.3 Deep ISO Minutiae Indexing (DIMI)

### 16.3.1 Processing Fingerprint Minutiae with CNNs

Fingerprint minutiae are a wide-spread representation of the fingerprint as biometric trait in biometric systems. The set of standardized features of a single minutia is typically reduced to only a few data fields, e.g. its position, angle and type. A single minutia on its own is not informative. Only sets of minutiae allow to use



| | | |
|---|---|---|
| $\Delta x1$ | $\Delta y1$ | ⋯ |
| $\Delta x2$ | $\Delta y2$ | ⋯ |
| ⋮ | ⋮ | ⋯ |
| $\Delta x$N | $\Delta yN$ | ⋯ |

**Figure 16.2:** Spatial and angular relations between an anchor minutia and its neighbours describe a neighbourhood. All neighbourhoods together yields a representation for an entire sample.

them for biometric recognition. Usually, spatial and angular relations between the minutiae enable the biometric recognition.

Our approach uses *neighbourhoods* of minutiae. A neighbourhood consists of an *anchor minutia* and its *neighbours*. $N$ nearest neighbouring minutiae are identified for each anchor minutia (see Figure 16.1(a)). In our approach, we found empirically $N = 13$ neighbours for description of a neighbourhood to be a good value. The more neighbours are used, the more computational effort is to be expected during index generation time but also the index can be expected to be more descriptive.

Each neighbourhood is aligned to the direction of the anchor minutia first. This enables rotational invariance. Then, features can be extracted from the relation of each neighbour to the anchor minutia (see Figure 16.3). In our approach we used Cartesian and Polar coordinates between an anchor and a neighbouring minutia as a feature. In addition, we added the angle of the aligned neighbouring minutia as a feature. All angular features were represented in a vectorized form. This allowed to extract seven features in total for each aligned neighbouring minutia with respect to an anchor minutia (see Table 16.1). Using these features allows to apply our FI approach to fingerprint systems, which are already rolled out in the field. However, extension of this approach is straight forward: Additional features per neighbour can just be appended to the existing ones.

Each fingerprint sample shall finally be represented by a single global representation. The global representation shall be assembled from a set of local represent-

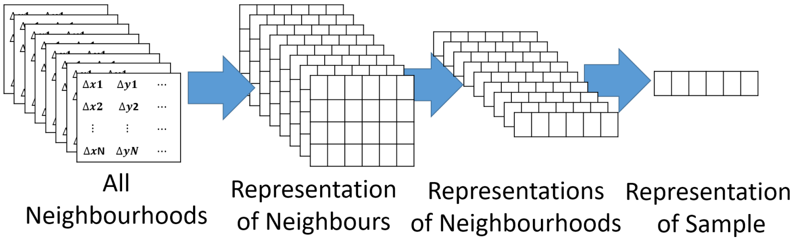


**Figure 16.3:** The first stage describes relations between each anchor minutia and its neighbours. Representations of neighbourhoods in the next stage yields a final representation for the entire sample.

| Nr. | Feature |
|---|---|
| 1 | Relative offset of the neighbour in x direction |
| 2 | Relative offset of the neighbour in y direction |
| 3 | Cosine part of angle difference between both minutiae |
| 4 | Sine part of angle difference between both minutiae |
| 5 | Distance between both minutiae |
| 6 | Cosine part of polar coordinate angle |
| 7 | Sine part of polar coordinate angle |

**Table 16.1:** Features used for representation of a relation between an anchor minutia and a neighbouring minutia.

ations, i.e. neighbourhoods of minutiae. Each neighbourhood is represented by features, which describe the relation of the anchor minutia to its $N$ neighbours.

To deal with the variable number of inputs we pick up an idea from *PointNet* [236]. PointNet was designed to classify point clouds. A CNN generates features for each point. At some stage of the CNN, maximum pooling is applied to all features over all samples. This yields permutation invariance. PointNet actually represents only single points. *PointNet++* extends this idea to work on pairs of points [237].

We propose to use the maximum pooling at two stages (see Figure 16.3 and Table 16.2). There is no natural order in the neighbouring minutiae. Order by distance to the anchor minutia will vary, if positional noise changes the distances of neighbours. Order can also fail entirely, e.g. if minutiae are missed by biometric feature extraction. To learn order invariant representations, we therefore apply a maximum over the dimension representing the neighbours in the first stage. In the second stage we want to enable invariance to the number of input minutiae. Applying the maximum function over the neighbourhoods yields representations for entire samples. Thus, this sample representations are finally invariant to the order of neighbours and invariant to the number of neighbourhoods. Cosine distance between the representations will be used as the similarity metric.

### 16.3.2 Architecture

The input to our model is a sample of dimensions (#Minutiae × #Neighbours × #Features) (see Figure 16.2). The actual number of neighbourhoods for a given sample depends on its number of minutiae. Despite this variability, CNNs need fixed numbers of dimensions as input. MaxNeighbourhoods may therefore be larger than the actual number of neighbourhoods in each sample. If the number of actual neighbourhoods is smaller, some space in the input representation of the sample will remain empty. Dedicated *Masking* layers (Masking) and pooling operations will take care of this mismatch. The Masking will leave out the empty neighbourhoods. By doing so, the empty neighbourhoods do not harm the entire processing.

The entire model may be divided into different stages. Each stage can be further subdivided into blocks. In the first two stages a block consists of a sequence of a *Convolutional* layer (ConvLayer), a *Batch Normalization* layer (BatchNorm), a *Parametric Rectified Linear Unit* layer (PReLU), and a Masking layer. The third stage consists of block of ConvLayers and PReLUs. Table 16.2 gives a systematic overview on the stages and their layers. Each stage works on different features (see Figure 16.3). The first stage works on the features of neighbouring minutiae. MaxPooling over the neighbouring minutiae yields a representation for

| Stage | Block | Type |
|---|---|---|
| Input | | Features<br>(Minutiae×13×7) |
| Neighbours | 1st | Conv (256×1×1×7)<br>Batch-Norm<br>Masking<br>PreLU |
| | 2nd | Conv (256×1×1×256)<br>Batch-Norm<br>Masking<br>PreLU |
| | 3rd | Conv (256×1×1×256)<br>Batch-Norm<br>Masking<br>PreLU |
| | | MaxPooling (1×13×1) |
| Neighbourhoods | 1st | Conv (128×1×1×256)<br>Batch-Norm<br>Masking<br>PreLU |
| | 2nd | Conv (128×1×1×128)<br>Batch-Norm<br>Masking<br>PreLU |
| | | MaxPooling (Minutiae×1×1) |
| Samples | 1st | Conv (64×1×1×128)<br>PreLU |
| | 2nd | Conv (64×1×1×64)<br>Flatten |
| Index | | Dense (16×64)<br>Spherical |

**Table 16.2:** DIMI consists of four stages: The first works on features of neighbouring minutiae, the second on neighbourhoods, and the third on entire samples. The last stage generates the fingerprint indices. MaxPooling between the stages enables necessary invariance.

neighbourhoods. Therefore, the second stage works on features for neighbourhoods. MaxPooling over the minutiae neighbourhoods yields a representation for entire samples. Those representations for samples are processed in the third stage. The final stage generates the feature indices by using a *Dense* layer (Dense) and a spherical projection.

The model was designed and trained using the framework *Tensorflow* [1]. The CNN contains less then 200k trainable parameters.

### 16.3.3 Training

The approach was trained entirely on a sequestered dataset "SQ FDB-35TC10Fx5" (Sonateq) of fingerprint samples provided by company Sonateq. The entire dataset consists of fingerprints from 3,500 individuals. Each individual provided all ten fingerprints with 5 samples each. The set therefore contains a total of 175,000 fingerprint samples. All fingerprint samples were acquired with optical livescanners with a resolution of $500\,\mathrm{dpi}$. The dataset was split into two parts: a training set and a validation set.[1] $75\%$ of the fingerprint samples were used for training. The remaining $25\%$ were used as a validation set.

We also did experiments on other datasets. We found, that large amounts of training data are necessary to achieve reasonable generalization capabilities of the trained CNN. Publicly available dataset therefore were simply too small for training and resulted in insufficient generalization. Experiments with synthetic fingerprints generated by tools SFinGe [46] or Anguli[7] did not achieve reasonable generalization to real fingerprint samples.[2] Fingerprint Minutiae were extracted by biometric feature extractor. We used three feature extractors in total: publicly available feature extractors MINDTCT[309] provided by NIST and FingerJetFX[65] provided by company Digital Persona and a COTS feature extractor.

The model was trained with the optimizer $Adam$[157]. There were no significant differences when using optimizers $Adagrad$ and $SGD$. Learning rate was set to $\lambda = 10^{-4}$. The number of samples per batch was selected to be 256.[3] A single batch consisted of 64 fingerprints with 4 samples each.

[1] In early experiments, we found, that there was no significant deviation in performance on a validation set and test set. We decided therefore to use only a validation set and to use as many samples as possible for training, since the amount of training data needed to be as large as possible.

[2] This is due to significant differences in the distributions of the input features between synthetic training data and real world test data. The largest difference was found for the distributions of angular difference between center and neighbouring minutiae. Synthetic neighbouring minutiae tend to have more similar orientations, while real minutiae orientations differ more.

[3] Larger batch sizes did not improve performance. Smaller batches resulted in instabilities during training.

The cost function makes use of the structure of a single training batch. Let a batch $B = \{s_k^m : k \in [1,..64], m \in [1,2,3,4]\}$ consist of fingerprint samples $s_k^m$ representing the $m$-th sample of the $k$-th fingerprint in the batch. Let CNN model $\mathcal{M}$ generate index vector $\mathcal{M}(s_k^m) \in \mathbb{R}^{16}$. The we can define the average index vector $\mu_k$ all samples of the $k$-th fingerprint as follows:

$$\mu_k = \frac{1}{4}\sum_{i=1}^{4} \mathcal{M}(s_k^i) \tag{16.1}$$

The cost function consists of two parts:

$$\text{cost} = \underbrace{\sum_{i=1}^{64}\sum_{j=0}^{4} |\sphericalangle(\mathcal{M}(s_i^j), \mu_i)|}_{\text{intra class}} - \underbrace{\sum_{i=1}^{64}\sum_{j=1}^{64} |\sphericalangle(\mu_i, \mu_j)|}_{\text{inter class}} \tag{16.2}$$

This cost function enforces two aspects. First, the CNN generates index vectors that are very similar for samples, which belong to the same fingerprint. Second, the average index vectors for all samples are spread over the entire unit sphere. Both aspects enable to train a reasonable indexing. This training strategy achieved significantly better results than training as a *Siamese Network*[34] or using *Triplet loss*[255].

Training was stopped, when the improvement on the validation set stopped. Training took only a few hours on a GPU[4].

### 16.3.4 Data augmentation

Generalization capabilities of a CNN can be improved by data augmentation. All augmentation works directly on minutiae and imitates typical challenges (see Figure 16.4). It contains several separate processes:

- *Spurious minutiae*: During biometric feature extraction sometime minutiae are found which do not exist. We added minutiae at random positions. The numbers of spuriously added minutiae were sampled from a distribution $P_{\text{spurious}} \sim \text{Poisson}\,(\lambda = 5.0)$. *Poisson* distribution are commonly used for description of processes of occurrences and processes of failures. Thus, a mean of five extra minutiae have been added to each sample.

- *Missing minutiae*: Sometimes biometric feature extraction misses existing minutiae. Data augmentation excluded randomly picked minutiae from the original set on minutiae. The numbers of excluded minutiae were sampled from a distribution $P_{\text{missed}} \sim \text{Poisson}\,(\lambda = 5.0)$.

[4] NVIDIA GTX1080 Ti

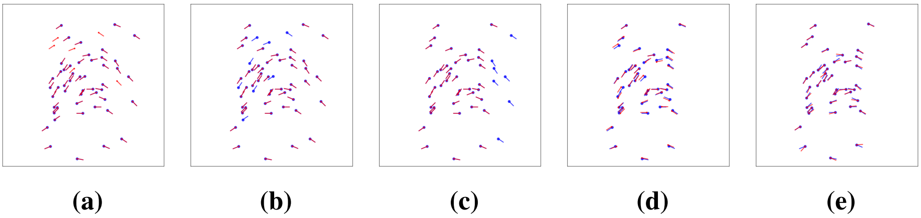


**Figure 16.4:** [
Augmentations for DIMI]Data augmentation imitates typical challenges: spurious minutiae (a), missing minutiae (b), partial fingerprints (c), positional inaccuracies (d), and angular inaccuracies (e). Minutiae before ●- and after ●- augmentation.

- *Positional noise*: Biometric feature extraction is not always precise in terms of position. Positional disturbances can also arise from deformations of the finger during the acquisition process. Noise is usually modeled by *Gaussian* distributions. Minutia positions have been slightly translated by noise distributed equal to a *Gaussian* distribution $\mathcal{N}_{0,0.02}$.

- *Angular noise*: Angular disturbances may arise from the acquisition process and imprecise feature extraction. Minutia angles have been augmented by noise distributed equally to a *Gaussian* distribution $\mathcal{N}_{0,0.2}$. This equals roughly a variance of $10°$.

- *Partial fingerprint*: The acquired fingerprint area usually varies between samples. Data augmentation imitated this behavior by excluding outmost minutiae in a terms of a random direction. While *missing minutiae* are randomly sampled from all minutiae, this augmentation has a spatial aspect. The percentages of excluded minutiae were equally distributed between $0$ and $20\%$: $\mathcal{U}_{[0,0.2]}$.

In addition, all training samples were also flipped. This doubles the amount of training data. Our approach is invariant to rotation and translation by design. Simple rotational and affine translational augmentation is therefore irrelevant. All augmentations but *angular noise* have impact on the nearest neighbours selection. This improves robustness to such variations. Data Augmentation is only applied to samples during training.

## 16.4 Experiments

### 16.4.1 Data

The central aspect of our experiments was the evaluation at the independent benchmark framework FVC-ongoing. FVC-ongoing has a benchmark called *Fingerprint Indexing*.[5] This benchmark consists of two parts: *FIDX-10K-1.0* (FIDX-10K) and *FIDX-50K-1.0* (FIDX-50K). Both benchmarks use datasets containing fingerprint samples acquired with optical live-scanners with a resolution of 500 dpi. Test FIDX-10K contains 10,000 fingerprint samples to be indexed and 100 fingerprint samples to be searched. Test FIDX-50K contains 50,000 fingerprint samples to be indexed and 500 fingerprint samples to be searched.

For further tests, we used those three datasets for testing, which are most commonly used for FI. All three datasets are publicly available: FVC 2000DB2[200], FVC2002DB1[201], and NIST SD04[307]. All images were acquired at a resolution of 500 dpi. The former two sets contain 100 finger with 8 impressions acquired with fingerprint live scanners. NIST SD 04 contains 2,000 rolled fingerprints with two impressions each. A common protocol is to build a gallery from the first impressions of all samples and use all other impressions as queries. The size of the test data is relatively small. Nevertheless, all datasets are commonly used for evaluation of FI approaches. Thus, they allow comparison with many approaches proposed in literature.

The fingerprint minutiae were extracted with a COTS biometric feature extractor. Input features for the CNN were generated from these fingerprint minutiae.

### 16.4.2 Metrics

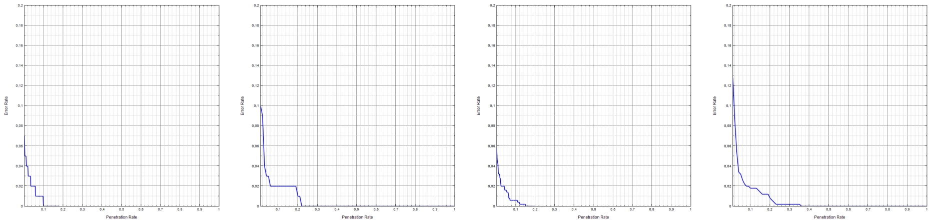
There are two metrics in terms of FI, which are somehow de facto standard: *penetration rate* (PR) and *error rate* (ER). The former is the relation between the count of candidates returned by the FI and the count of indexing samples. Therefore, it represents the percentage of the data, which remains in the candidate list after FI. The later rate represents the rate of mated samples which are missed by the returned list of candidates. Those mated samples cannot be found by the thorough comparison stage later. Both rates depend on how a candidate list is created. For these experiments a sample is included into the candidate list, if the distance is smaller than a given threshold.

In addition, we report the rate *IS* for an incremental search retrieval scenario [52]. Each candidate list is of exactly the appropriate size to include the mated sample. The rate IS is the mean length of such an ideal candidate list. Even-though this is

[5] Requires a `win32` console application conform with a test protocol.

| Approach | FIDX-10K | | | FIDX-50K | | |
|---|---|---|---|---|---|---|
| | **ER$_{100}$** | **ER$_{1000}$** | **IS** | **ER$_{100}$** | **ER$_{1000}$** | **IS** |
| MCC | 5.898% | 9.888% | 1.215% | 6.450% | 15.140% | 0.954% |
| *DIMI*$_{\text{COTS}}$ | 20.0% | 22.0% | 0.723% | 20.0% | 36.0% | 0.747% |

**Table 16.3:** Results at FVC-ongoing. Rates ER$_{100}$ is the benchmark's definition for PR@ER=0.01 and ER$_{1000}$ for PR@ER=0.001 respectively.



**(a)** MCC on FIDX-10K **(b)** DIMI on FIDX-10K **(c)** MCC on FIDX-50K **(d)** DIMI on FIDX-50K

**Figure 16.5:** *Cumulative Match Characteristics* (CMC) visualize the tradeoff between Penetration rate and Error Rate. They allow comparison of MCC and DIMI. Plots are provided by FVC-ongoing.

an idealized approach, it allows to estimate an ideal penetration rate independent from any candidate list construction policy. It is also independent from any errors made by the thorough comparison of the fingerprints in the candidate list. This focus on FI makes IS a well suited metric.

### 16.4.3 Results

The results achieved at the independent benchmark framework FVC-ongoing are of most interest. About two hundred algorithms or rather versions of algorithms were evaluated at FVC-ongoing so far. [6] Unfortunately, only for MMC results were published. Table 16.3 allows comparison of MCC and our approach DIMI. An IS rate of 0.747% was achieved for test FIDX-50K and 0.723% for FIDX-10K respectively. DIMI therefore outperforms the only other evaluated method MCC in this metric. MCC achieved 0.954% in this metric for FIDX-50K and 1.215% for FIDX-10K respectively. DIMI improves the performance by 22% and 40% respectively compared to MCC. However, in both other metrics DIMI performs worse than MCC. Figure 16.5 visualizes the corresponding *Cumulative*

[6] https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/Statistics.aspx

*Match Characteristics* (CMC) curves for MCC and DIMI for both parts of the benchmark.

Table 16.4 summarizes the results achieved with our approach on publicly available datasets. We found that, DIMI performs significantly better when using COTS feature extractor compared to using MINDTCT and FingerJetFX feature extractors. It also allows comparison with other minutia-based approaches which reported results for the datasets. There were slight differences in the performance between training and validation data. DIMI performs extremely well on the validation set, which was split from the dataset Sonateq. When applied to other datasets DIMI performs worse: 3.5% for FVC2000DB2, 4.0% for FVC2002DB1, and 7.9% for NISTSD04 (see also Table 16.4). The performance for NISTSD04 was significantly worse. This may be due to the differences in characteristics of rolled fingerprint samples and plain fingerprint samples. Unfortunately, comparability with other approaches is limited, since there are no unified processes for reporting indexing performances: Each approach evaluates with different metrics on different data.[7] In some cases metrics must be estimated by reading graphs (indicated in the Table by $\sim$).

Interoperability is an important issue in biometric recognition. Table 16.5 lists IS rates for models, which were trained on features extracted with a specific feature extractor but tested on minutiae extracted with a different extractor. The minutiae extracted with FingerJetFX and COTS seem to enable interoperability.

We also tested on a set of additional impressions of the fingerprint samples used for training. The difference in performance on those impression and the test data is small, too. This indicates, that our approach is not explicitly over-fitting to the specific samples used during training.

The proposed indexing comes with additional costs in terms of memory resources. For instance, an ISO/IEC template generated by minutiae extractor FingerJetFX has a mean size of about 336 Bytes per sample of FVC2000 DB2. The proposed index feature vector is of size 16 float values. This equals 64 Bytes. MCC uses 6.4 KB in average per sample [52]. Adding the indexing feature vector to the template increases the required amount of data by about 19% for samples of FVC2000 DB2.

## 16.5 Conclusion

We presented an approach called *Deep ISO Minutiae Indexing* (DIMI) to generate a FI vector from neighbourhoods of minutiae. Only standard ISO/IEC minutia

[7] Drozdowski et al. proposed metrics for relevant performances [69].

| **Dataset** | **Approach** | | **ER** | **ER** |
|---|---|---|---|---|
| | | **IS** | **@PR=0.05** | **@PR=0.01** |
| Sonateq (Train) | $DIMI_{\text{COTS}}$ | 0.0023 | 0.0030 | 0.0320 |
| | $DIMI_{\text{MD}}$ | 0.0059 | 0.0172 | 0.1490 |
| | $DIMI_{\text{FJ}}$ | 0.0031 | 0.0065 | 0.0720 |
| Sonateq (Validation) | $DIMI_{\text{COTS}}$ | 0.0038 | 0.0110 | 0.0540 |
| | $DIMI_{\text{MD}}$ | 0.0077 | 0.0281 | 0.1753 |
| | $DIMI_{\text{FJ}}$ | 0.0049 | 0.0158 | 0.1042 |
| FVC2000DB2 | $DIMI_{\text{COTS}}$ | 0.0354 | 0.1686 | 0.4671 |
| | $DIMI_{\text{MD}}$ | 0.0608 | 0.3286 | 0.6814 |
| | $DIMI_{\text{FJ}}$ | 0.0439 | 0.2043 | 0.5286 |
| | [52] | 0.0172 | $\sim$0.03 | $\sim$0.056 |
| | [333] | n/a | 0.09 | 0.12 |
| | [155] | n/a | $\sim$0.07 | $\sim$0.09 |
| | [156] | n/a | $\sim$ 0.08 | $\sim$ 0.06 |
| | [79] | n/a | $\sim$ 0.04 | $\sim$ 0.025 |
| | [11] | n/a | $\sim$ 0.18 | $\sim$ 0.05 |
| FVC2002DB1 | $DIMI_{\text{COTS}}$ | 0.0399 | 0.1660 | 0.4249 |
| | $DIMI_{\text{MD}}$ | 0.0640 | 0.2800 | 0.5971 |
| | $DIMI_{\text{FJ}}$ | 0.0508 | 0.2257 | 0.5271 |
| | [52] | 0.0137 | $\sim$0.012 | $\sim$0.02 |
| | [333] | n/a | 0.06 | 0.09 |
| | [297] | n/a | $\sim$ 0.13 | $\sim$ 0.42 |
| | [178] | n/a | $\sim$ 0.10 | $\sim$ 0.18 |
| | [106] | 0.1031 | $\sim$ 0.21 | $\sim$ 0.36 |
| | [302] | n/a | $\sim$ 0.12 | $\sim$ 0.34 |
| | [183] | n/a | $\sim$ 0.02 | $\sim$ 0.09 |
| | [211] | n/a | $\sim$ 0.01 | $\sim$ 0.008 |
| | [156] | n/a | $\sim$ 0.04 | $\sim$ 0.03 |
| | [79] | n/a | $\sim$ 0.01 | $\sim$ 0.05 |
| | [114] | n/a | $\sim$ 0.02 | n/a |
| | [303] | n/a | $\sim$ 0.02 | $\sim$ 0.01 |
| | [10] | n/a | $\sim$ 0.4 | $\sim$ 0.05 |
| | [11] | n/a | $\sim$ 0.07 | $\sim$ 0.02 |
| | [78] | n/a | $\sim$ 0.17 | $\sim$ 0.1 |
| | [9] | n/a | $\sim$ 0.08 | $\sim$ 0.0 |
| NIST SD04 | $DIMI_{\text{COTS}}$ | 0.0785 | 0.3280 | 0.5445 |
| | $DIMI_{\text{MD}}$ | 0.1057 | 0.4550 | 0.7000 |
| | $DIMI_{\text{FJ}}$ | 0.1006 | 0.3850 | 0.6245 |
| | [52] | 0.0159 | $\sim$0.046 | $\sim$0.067 |
| | [212] | n/a | $\sim$0.036 | $\sim$ 0.058 |
| | [25] | n/a | $\sim$0.17 | $\sim$0.23 |
| | [155] | n/a | $\sim$0.07 | $\sim$0.085 |
| | [211] | n/a | $\sim$ 0.08 | $\sim$ 0.03 |
| | [156] | n/a | $\sim$ 0.08 | $\sim$ 0.07 |
| | [79] | n/a | $\sim$ 0.11 | $\sim$ 0.05 |
| | [11] | n/a | $\sim$ 0.25 | $\sim$ 0.06 |
| | [11] | n/a | $\sim$ 0.09 | $\sim$ 0.06 |
| | [9] | n/a | $\sim$ 0.08 | $\sim$ 0.05 |

**Table 16.4:** Results on other test data. Our approach performs well, when test and training data are similar.

| | | Test minutiae | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | FVC2000 DB2 | | | FVC2002 DB1 | | | NIST SD04 | | |
| | | COTS | MD | FJ | COTS | MD | FJ | COTS | MD | FJ |
| Train minutiae | COTS | 0.0354 | 0.2258 | 0.0439 | 0.0399 | 0.2157 | 0.0499 | 0.0785 | 0.2182 | 0.0853 |
| | MD | 0.1853 | 0.0608 | 0.2290 | 0.1829 | 0.0640 | 0.2061 | 0.2305 | 0.1057 | 0.2259 |
| | FJ | 0.0457 | 0.2242 | 0.0439 | 0.0542 | 0.2206 | 0.0508 | 0.1031 | 0.2138 | 0.1006 |

**Table 16.5:** IS rates reveals aspects on interoperability. Compatibility between different feature extractors is important. Models trained on minutiae extracted with the COTS feature extractor perform similar with features extracted with FingerJetFX (FJ) and vice versa. However, when those models use minutiae extracted with MINDTCT (MC) performance decays significantly.

templates are used as input features, which allows compatibility with biometric recognition systems in which only minutiae but no fingerprint samples are available. The approach is easily extendable to additional features. Our main contribution is to allow CNNs to deal with variable number of input minutiae in any order by systematic use of the maximum function. This allows invariance to the order of neighbours in each neighbourhood of minutiae and invariance to the number of neighbourhoods. Index vectors are very short and allow very fast comparison based on only 16 float values. An IS rate of 0.747% for test FIDX-50K and 0.723% for FIDX-10K respectively at independent benchmark framework FVC-ongoing was achieved. These are the best reported results for so far.

# Chapter 17

# Estimating the Data Origin of Fingerprint Samples

**Summary** Fingerprint samples vary significantly in their appearance. The appearance depends on the data origin. Knowledge on the data origin allows specialized processing. Six method of classification were evaluated on fifteen datasets for their capabilities of estimating the data origin. Estimation can be performed very accurate and reliable.
This publication is joint work with Jan Marek May and Christoph Busch. It was presented at the *BIOSIG* in Darmstadt (Germany) in 2018.

[262] P. Schuch, J. M. May, and C. Busch. Estimating the data origin of fingerprint samples. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553235

**Abstract** The data origin (i.e. acquisition technique and acquisition mode) can have a significant impact on the appearance and characteristics of a fingerprint sample. This dataset bias might be challenging for processes like biometric feature extraction. Much effort can be put into data normalization or into processes able to deal with almost any input data. The performance of the former might suffer from this general applicability. The latter losses information by definition. If one is able to reliably identify the data origin of fingerprints, one will be able to dispatch the samples to specialized processes. Six methods of classification are evaluated for their capabilities to distinguish between fifteen different datasets. Acquisition technique and acquisition mode can be classified very accurately. Also, most of the datasets can be distinguished reliably.

## 17.1 Introduction

No two fingerprints are the same. Every fingerprint is at least slightly different from another. This fact makes a fingerprint a valuable trait for biometric recognition. However, two fingerprint samples of the very same fingerprint can also be very different. An important source of variation arises from the two aspects of a sample's *data origin*: *acquisition technique* and *acquisition mode*.

Fingerprint samples might have been acquired with different techniques. For example, they can be acquired using dedicated fingerprint livescanners. There is a variety of manufacturers and devices. The latter can differ in physical acquisition principles, e.g. optical or capacitive. Besides dedicated devices, almost any camera may be used for acquisition. Fingerprints may also be acquired using *ink* and paper. There are also *latent* fingerprints or *fingermarks*, which are typically evidences from crime-scenes. Those fingerprints are captured using special techniques, e.g. photography of fingerprints highlighted with powders.

Not only the technique of acquisition matters, but also *how* the finger is presented. There are namely four modes of acquisition. The finger may be placed *plain* on an acquisition surface. Fingerprints can be *rolled* over an acquisition surface. Some devices require the fingerprint to be *swiped* over a line scanner. While these three modes are all contact-based, the fingerprint may also be acquired *contact-less*. Many of the possible combinations of technique and mode are actually deployed in the operational scenarios.

In general, the *fidelity* of a fingerprint sample to its source depends on the data origin [123]. Thus, all data is *biased* by its origin. This *dataset bias* tends to be very different among different datasets. Dealing with such differences can be challenging for any process using the data. And it turns out that biometric comparison

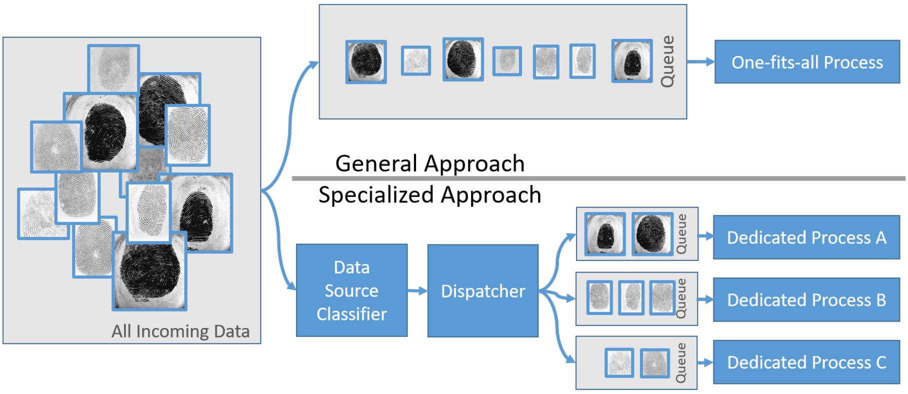


**Figure 17.1:** Knowing a sample's origin enables usage of specialized processing methods.

across different data origins is challenging [140].

There are two ways of dealing with the dataset bias (see Figure 17.1). *One-fits-all solutions* need to be able to deal with any input. This general applicability comes at the cost of recognition performance. *Dedicated* or *specialized modules* modules can be tailored to the special needs of an input at hand. The more the inputs differ, the larger is the benefit of a specialized processing pipeline. If you do not know, what you are processing, you will have to apply a one-fits-all solutions. But if you do know, what kind of data you are processing, you will be able to benefit from specialization. There are standardized data interchange formats, which provide meta information about acquisition mode and technique, e.g. the international standard ISO/IEC 19794-4 [127]. But fingerprint samples do not always come with such information about their origin or the information is not reliable. This lack of information may be unintentional, e.g. when processing legacy data in a system, in which the samples have various data origins. Or it may be by intention, e.g. in benchmarks or competitions like FVC-ongoing [68]. If you still want to apply specialized approaches, you will have to guess the origin of the input data. Guessing the origin essentially makes use of dataset bias, since this bias is what makes data distinguishable. In this case, the dataset bias is a desirable property.

Ghiani et al. proposed linear *Support Vector Machines* for pairwise discrimination of datasets [88]. This is the first extensive investigation of methods for estimating the origin of fingerprint samples using a multitude of datasets. In this work, we propose to use a *Convolutional Neural Network* (CNN) for guessing the origin of the fingerprint. To assess this approach, its performance is then compared to other prominent classifiers.

The rest of this work is organized as follows: Section 17.2 gives an overview on related research. A CNN based approach is described in Section 17.3. Section

17.4 describes the experiments with this approach and five alternative classifiers. It also presents the results achieved. A conclusion can be found in Section 17.5.

## 17.2 Related work

Capabilities of generalization are always important for methods of pattern recognition. If a pattern recognition method is not able to deal with other data than the data used for training, the method will be useless. The dataset bias, also known as the *covariate shift*, might be the most challenging aspect for generalization. Dealing with the dataset bias is therefore an important topic in pattern recognition. A lot of research has been done in enabling methods to deal with data, that has not been seen during the training phase.

Besides the improvements in the generalization capabilities, there is another way to look at the dataset bias. For example, Torralba and Efros investigated how well image datasets can be distinguished [287]. They evaluated their method on datasets available at the time for large classification benchmarks. They called this task the *Name the Dataset* game. Their motivation was to measure and to understand the bias in datasets. This bias may usually result in generalization problems for classifiers, when the test data differs from the training data.

The challenge of generalization becomes even more important as the accuracies rise. In case of high accuracies, slight differences in performances between training and test data can result in significant relative differences between these performances. CNNs are state of the art in several domains of image processing and pattern recognition. The accuracies achieved with CNNs are remarkably high and dataset biases can have a strong impact here. Tommasi et al. did extensive experiments on the dataset bias when using CNNs [286].

The dataset bias is a prominent challenge in fingerprint recognition. This challenge is also known as *cross device biometric recognition*. Jia et al. developed a dataset, which contains fingerprint samples from the same fingerprint acquired with nine different devices [140]. They showed that recognition across different devices is challenging.

Ghiani et al. proposed to use a linear *Support Vector Machine* for classification of different datasets [88]. They used the classification to apply specialized methods of presentation attack detection on the fingerprint samples. They evaluated classification between pairs of datasets. This work provides comparison of six methods of classification and evaluation on a multitude of fingerprint datasets.

## 17.3 CNNs for Data Origin Estimation

Estimating the data origin is a typical *classification* task. CNNs are currently state of the art in classification tasks. We therefore propose to use a CNN for this task.

**Table 17.1:** Straight forward architecture of proposed CNN.

| # | Layer | Output |
|---|---|---|
| 0 | Input (192×192×1) | (192×192×1) |
| 1 | ConvLayer (32×3×3×1) | (190×190×32) |
| 2 | MaxPooling (2×2) | (95×95×32) |
| 3 | PReLU | (95×95×32) |
| 4 | ConvLayer (32×3×3×32) | (93×93×32) |
| 5 | MaxPooling (2×2) | (46×46×32) |
| 6 | PReLU | (46×46×32) |
| 7 | ConvLayer (32×3×3×32) | (44×44×32) |
| 8 | MaxPooling (2×2) | (22×22×32) |
| 9 | PReLU | (22×22×32) |
| 10 | ConvLayer (32×3×3×32) | (20×20×32) |
| 11 | MaxPooling (2×2) | (10×10×32) |
| 12 | PReLU | (10×10×32) |
| 13 | ConvLayer (32×3×3×32) | (8×8×32) |
| 14 | MaxPooling (2×2) | (4×4×32) |
| 15 | PReLU | (4×4×32) |
| 16 | Flatten | (512) |
| 17 | Dense (32×512) | (32) |
| 18 | PReLU | (32) |
| 19 | Dense (32×32) | (32) |
| 20 | PReLU | (32) |
| 21 | Dense (15×32) | (15) |
| 22 | Softmax | (15) |

In the following we will sketch an architecture for a CNN, which is capable of this task. We will also provide information on how the training of the CNN was performed.

### 17.3.1 Architecture

Table 17.1 gives an overview over the entire architecture and the outputs of each layer. There has been no extensive optimization of any hyperparameters. The entire model is built from six different types of layers: *Convolutional* layers (ConvLayer), *Parametric Rectified Linear Units* (PReLu), *Maximum Pooling* layers (MaxPooling), *Flatten* layers (Flatten), *Dense* layers (Dense), and *Softmax* layers (Softmax). All ConvLayer have 32 kernels. All kernels are 3x3 filters. There was no *striding* in the ConvLayers. MaxPooling was performed on 2x2 blocks. The CNN has less than 60,000 trainable parameters.

Generalization is always a crucial issue when training a classifier. In this approach, the capability of generalization was enforced by adding a $l_2$-regularization on the kernels of the Dense layers. A common approach to strengthen the generalization capabilities of a CNN is the introduction of *Batch Normalization* [116]. However, usage of Batch Normalization in this approach led to an undesired behavior: The CNN was not able to distinguish between the different datasets any more.

The input to the CNN is a grey scale image, which is cropped to its central region of size 192x192 pixels (see Figure 17.2). The approach does not rely on any foreground detection. It simply crops a region of interest from the center. This allows an automatic processing. The CNN therefore does only see a small part of the fingerprint sample.

The output of the Softmax layer can be understood as the likelihood for the fifteen respective classes of datasets (see Section 17.4.1). The model was created and trained in the Deep Learning (DL) framework *Tensorflow* [1].

### 17.3.2 Training

The model was trained with the optimizer $Adam$[157] . There were no significant differences when using optimizers $Adagrad$ or $SGD$. Learning rate was set to $\lambda = 10^{-4}$.

The number of samples per batch was selected to be 128. Larger batch sizes did not improve performance. Smaller batch sizes resulted in instabilities during training. The network was trained to minimize cross validation loss. The samples in each batch were randomly picked from the training data. *Data augmentation* was applied to the fingerprint samples. The samples were rotated randomly. The 192x192 pixel region used as the input was cropped randomly from a region nearby the sample's center. Such a data augmentation is commonly used to increase the amount of training data. This indirectly prevents the CNN from overfitting to the training data. Therefore it also helps enabling generalization of the CNN.

For training the CNN, the training set was split into two parts. About two thirds of the fingerprint samples were used as training data. The remaining fingerprint samples were used as a validation set. Training was stopped when the improvement of loss for the validation set stopped. Such an *early stopping* strategy is a common method to prevent a CNN from over-fiting. Training the CNN took less than an hour on a GPU[1].

## 17.4 Experiments

### 17.4.1 Datasets

The fifteen tested datasets are a subset of publicly available datasets. Only publicly available dataset were chosen to allow reproducibility of the results. Even though, some dataset might be some kind of out-dated, they still represent legacy data.

[1] NVIDIA GTX 780

**Table 17.2:** Datasets used for evaluation.

| # | Dataset | Acquisition Technique | Acquisition Mode | Ref |
|---|---|---|---|---|
| 1 | FVC2000 DB1 | Optical | Plain | [200] |
| 2 | FVC2000 DB2 | Capacitive | Plain | |
| 3 | FVC2000 DB3 | Optical | Plain | |
| 4 | FVC2002 DB1 | Optical | Plain | [201] |
| 5 | FVC2002 DB2 | Optical | Plain | |
| 6 | FVC2002 DB3 | Capacitive | Plain | |
| 7 | FVC2004 DB1 | Optical | Plain | [202] |
| 8 | FVC2004 DB2 | Optical | Plain | |
| 9 | FVC2004 DB3 | Thermal | Swiped | |
| 10 | FVC2006 DB2 | Optical | Plain | [47] |
| 11 | FVC2006 DB3 | Thermal | Swiped | |
| 12 | MCYT DP | Optical | Plain | [228] |
| 13 | MCYT PB | Capacitive | Plain | |
| 14 | NIST DB4 | Ink-based | Rolled | [307] |
| 15 | NIST SD14 | Ink-based | Rolled | [306] |

Each dataset represents its own class for classification. Table 17.2 summarizes details of the single datasets used during our experiments. These datasets represent a subset of the variability of *acquisition techniques* and *acquisition modes*.[2] There are datasets, which were acquired by livescanners using optical, capacitive, and thermal sensors. There are also two datasets acquired by using ink-based techniques. Most of the datasets contain plain fingerprints. Two datasets contain rolled fingerprints and two datasets contain swiped fingerprints. Each data will represent one class in the classification. Thus, it will be a multi-class classification.

The number of samples in the datasets differs. Using all samples in the evaluation would have imbalanced the influence of each dataset. Therefore, only the first 800 samples in each dataset have been selected for these experiments. By doing so, all datasets have the same amount of training data and testing samples respectively. The images were cropped to their central region of 192x192 pixels. This prevents the classifiers to learn from trivial features like image dimensions or any systematic artifacts at the borders of the fingerprint samples.

[2] In terms of ISO/IEC 19794-4 the acquisition technique may be deducted from the *capture device ID* identifier. The acquisition mode is represented by the *impression type* identifier in the standard.

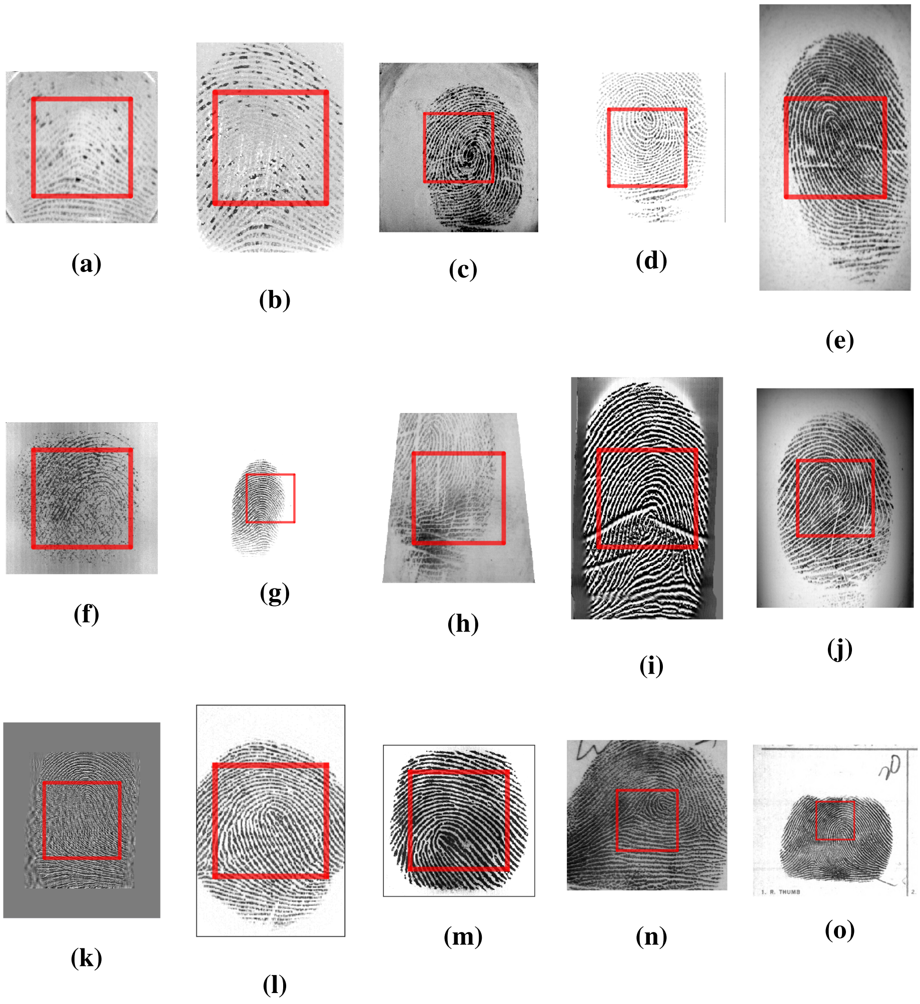


**Figure 17.2:** Samples from the datasets used for evaluation (see Table 17.2 for numbering). The red square indicates the 192x192 crop region used for training and testing.

### 17.4.2 Metrics

The performance of the classifiers was assessed by the *accuracy* acc in predicting the correct data origin. Three accuracies were evaluated. First, the accuracy $\text{acc}_{\text{dataset}}$ of estimating the correct dataset was evaluated. Second, the accuracy $\text{acc}_{\text{mode}}$ for estimating the correct acquisition mode was measured. Third, the accuracies $\text{acc}_{\text{tech}}$ for estimating the correct acquisition technique was measured.
Let a tuple $(x_i, y_i)$ contain the $i$-th fingerprint sample $x_i$ and its actual class $y_i \in Y$ in the set $Y$ of all classes. Let $F(x_i)$ be the estimated class for fingerprint sample $x_i$. The accuracy $\text{acc}_{\text{dataset}}$ for a set $X = \{x_i : i \in [1, N]\}$ containing $N$ samples is therefore the expected value for the rate of correct estimations for the dataset and can be calculated by using the indicator function $\mathbb{1}$:

$$\text{acc}_{\text{dataset}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F(x_i)=y_i} \tag{17.1}$$

Let $F_t(y) = m : y \in Y \mapsto t \in T$ be the function that maps a class $y$ to it corresponding mode $t$ in the set $T$ of all acquisition modes, i.e. $T = \{$'Optical', 'Capacitive', 'Thermal', 'Ink-based'$\}$. Then the accuracy $\text{acc}_{\text{tech}}$ can be calculated as follows:

$$\text{acc}_{\text{tech}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F_t(F(x_i))=F_t(y_i)} \tag{17.2}$$

This accuracy $\text{acc}_{\text{tech}}$ can be understood as the expected value for the rate of correct estimations for the acquisition technique.
Respectively, let $F_m(y) = m : y \in Y \mapsto m \in M$ be the function that maps a class $y$ to it corresponding mode $m$ in the set $M$ of all acquisition modes, i.e. $M = \{\text{'Plain'}, \text{'Rolled'}, \text{'Swiped'}\}$. Then the accuracy $\text{acc}_{\text{mode}}$ can be calculated as follows:

$$\text{acc}_{\text{mode}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F_m(F(x_i))=F_m(y_i)} \tag{17.3}$$

This accuracy $\text{acc}_{\text{mode}}$ can be understood as the expected value for the rate of correct estimations for the acquisition mode.

The priors of the acquisition modes and acquisition technique are not equally distributed over all classes. This imbalance has of course impact on the respective accuracy measures.

4-fold cross-validation was used to allow a more reliable evaluation. Each fingerprint dataset was therefore split into four parts of equal size. In each fold of the evaluation, one of the parts was kept out of the training data and used for testing only. No fingerprint sample is in more than one testing split. The splits were

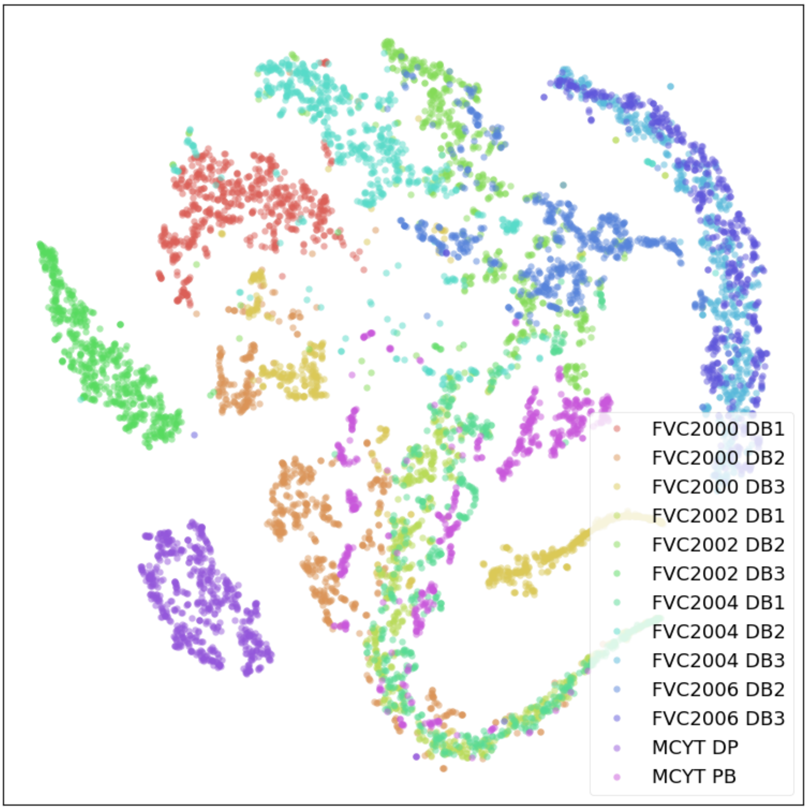


**Figure 17.3:** t-SNE embedding for GLCM features already allows distinguishing between some datasets.

performed randomly. All datasets have more than one fingerprint sample per fingerprint. It was enforced, that all samples of the same fingerprint are in the same split. Therefore no two samples belonging to the same finger are in the training split and the test split of the same fold. The fact, that fingerprint samples stemming from the same source (finger instance) were splitted into different datasets, was neglected. The accuracies reported here are actually the mean accuracies over all four evaluation runs.

For inspection of the failures in classifying the datasets, *confusion matrices* are calculated. Those matrices allow to analyze the failures made with respect to the real data origin.

### 17.4.3 Alternative Classifiers

Five alternative classifiers were tested to benchmark the performance of the CNN based approach: Random Forest Classifier [33], Extra Trees Classifier [86], Decision Tree Classifier, Logistic Regression and K Nearest Neighbor Classifier.[3] All alternatives have implementations in the python based machine learning toolbox *scikit-learn* and can be used out of the box [234].
Classification applied directly to the signal of the central crops did not perform well. Thus, *Grey level co-occurrence matrices* (GLCM) have been calculated for each central crop.[4] Those represent the entire range of grey level values in a crop and the dynamics of neighboring pixels [104]. The intensities of grey values were subsampled by a factor of 4 to reduce the number of features to a reasonable order.
A common step in classic pattern recognition is to do *Feature Selection*. *Principal Component Analysis* (PCA) is probably the most standard method here. PCA analyses input data for their components of maximal variance. PCA transforms input data to a new base. This base allows to select only those axes with the largest variance in the data. Figure 17.3 visualizes a two dimensional embedding of the reduced feature set, which was generated by *t-distributed Stochastic Neighbour Embedding* (t-SNE)[197]. Obviously, the input features allow distinguishing between most datasets.
All approaches were evaluated on the full set of input features and also on a reduced feature set. The reduced set contained the important components, which together explain more than $99.9\%$ of the variance in the original data. The PCA based transformation was calculated on the training data only, of course.

### 17.4.4 Results

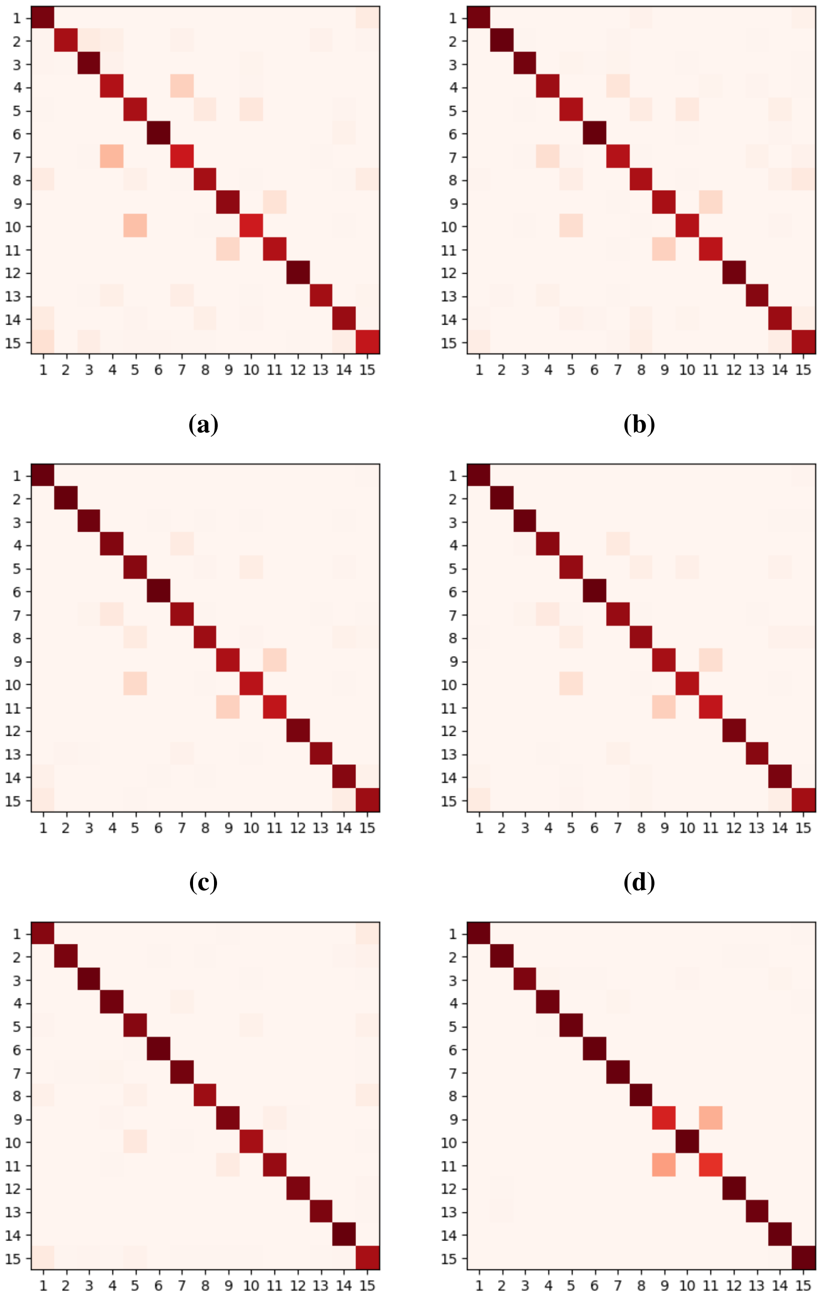
Table 17.3 summarizes the results of the evaluated methods. Figure 17.4 visualizes the confusion matrices for classifications based on the features derived from entire GLCMs.

Most of the classifiers were able to classify the acquisition mode and the acquisition technique very reliable. The CNN based approach achieved an accuracy of $99.7\%$ for estimating mode and $99.5\%$ for technique respectively. The best alternative classifier achieved similar accuracies.

Despite this, dataset classification is more challenging. Most of the classifiers are able to distinguish very reliably even between the different datasets. Of course,

[3] AdaBoost, Huber Regressor, and SVM were also tested but failed totally to learn a classification.

[4] GLCM is a classic feature for texture classification. Other features for texture classification may be applied, of course. However, GLCM already yielded impressive results for data origin classification.



(a)

(b)

(c)

(d)

(e)

(f)

**Figure 17.4:** The confusion matrices can be used to identify class-specific failures made by a classifier. Each dataset is identified by the numbering provided in Table 17.2. Most failures in classification are made between datasets 9 and 11 (both thermal swipe), between datasets 14 and 15 (both ink-based rolled), and datasets 5 and 10 (both likely the same scanner).

**Table 17.3:** The most relevant accuracies are those achieved on test data. Best result is marked in bold.

| Input | Classifier | Feature | $\textbf{acc}_{\textbf{dataset}}$ | | $\textbf{acc}_{\textbf{mode}}$ | | $\textbf{acc}_{\textbf{tech}}$ | |
|---|---|---|---|---|---|---|---|---|
| | | **Selection** | **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| GLCM | Decision Tree | none | 100% | 89.2% | 100% | 97.4% | 100% | 96.7% |
| | | PCA | 100% | 83.2% | 100% | 93.5% | 100% | 91.6% |
| | Extra Trees | none | 100% | 92.9% | 100% | 98.5% | 100% | 98.3% |
| | | PCA | 100% | 90.2% | 100% | 97.0% | 100% | 96.2% |
| | Random Forest | none | 99.8% | 93.2% | 100% | 98.6% | 100% | 98.4% |
| | | PCA | 99.8% | 91.1% | 99.9% | 97.0% | 99.9% | 96.4% |
| | Logistic Regression | none | 100% | **95.6%** | 100% | 98.1% | 100% | 97.9% |
| | | PCA | 84.3% | 83.3% | 94.9% | 94.5% | 93.6% | 93.0% |
| | K Nearest Neighbors | none | 92.8% | 86.7% | 98.1% | 96.6% | 97.4% | 95.2% |
| | | PCA | 92.6% | 86.4% | 98.0% | 96.5% | 97.3% | 95.1% |
| Images | CNN | none | 100% | 94.7% | 100% | **99.7%** | 100% | **99.5%** |

the most important aspect is the accuracy achieved for the test sets. The best result of all evaluated approaches is achieved by the Logistic Regression: 95.6% of the samples were classified correctly.

Some pairs of datasets were confused more often than others. Failures are made by confusing samples from the two thermal/swipe datasets. Samples from both datasets containing rolled fingerprints were also confused in some cases. Finally, samples from datasets FVC2002D DB2 and FVC2006 DB2 often were also misclassified. Actually, both datasets were acquired with scanners from the same manufacturer. It is likely, that the same scanner model or even the very same scanner was used for acquisition. If so, the classifiers based their decision on additional dataset biases, e.g. environmental influences.

All classifiers overfitted to the training data. All training data could be classified almost perfectly by all classifiers. However, using the PCA for feature selection did not work out well. In general, the accuracy achieved on the test data was lower, when feature selection was applied before classification.

## 17.5 Conclusion

The dataset bias can be a challenge for any process, which has to deal with unknown input data. We propose to exploit the database bias. If one can use the dataset bias as a distinguishing property for the origin of a fingerprint sample, one will be able to use this information to dispatch the sample to a process, which is specialized on such inputs. Six classifiers to guess the origin of a fingerprint were evaluated. Acquisition mode and acquisition technique were classified very reliable. Fifteen datasets containing their individual dataset biases were tested for evaluation. Most of the conventional classifiers worked well out of the box: Accuracy for the estimation was over 95%. The classification errors do not distribute equally among the different classes. While most of the datasets were distinguished reliably by the classifiers, some are harder to be distinguished. The CNN based approach and the conventional approaches performed similar.

## Acknowledgments

The authors thank the *15th IAPR/Eurasip International Summer School for Advanced Studies on Biometrics for Secure Authentication* for inspiration and all suppliers of datasets used in our experiments.

# Chapter 18

# Conclusion on Efficient Processing Structures

This Part IV is dedicated to answer the research question $\mathcal{RQ}_3$ *"Can DL provide solutions for efficient processing structures?"* There are many aspects in a fingerprint recognition system to improve on. The most central aspects here may be Fingerprint Indexing, as it has significant impact on biometric performance and speed. A survey provides an overview about a hundred different approaches for Fingerprint Indexing. DL-based approach DIMI performs competitive to state of the art approach MCC. DIMI even outperformed MCC at benchmark framework FVC-ongoing with respect to the metric IS representing the average candidate list length. Compared to MCC, DIMI reduces the IS rate from $1.21\%$ to $0.72\%$ for test FIDX-10K and from $0.95\%$ to $0.75\%$ for test FIDX-50K respectively. This allows to increase the throughput in the incremental search scenario, while keeping the recognition accuracy. DL can therefore be considered at a competitive alternative to hand-crafted approaches in Fingerprint Indexing.

CNNs are very versatile. Application of DL to further aspects in fingerprint recognition systems was demonstrated as well. They can be used to learn a rotational alignment as well as to estimate the data origin of fingerprint samples.

The findings for fingerprint alignment, data origin estimation, and especially for pre-selection algorithms, yield an enormous potential of Deep Learning Thus, it can be concluded with respect to the research question $\mathcal{RQ}_3$, that DL can provide manifold improvements on the efficient processing in fingerprint recognition systems.

# Part V

# Conclusion on Deep Learning for Fingerprint Recognition Systems

The research of this thesis had the following objective:

*"Comparison of hand-crafting and Deep Learning approaches in fingerprint recognition"*

This research frame is very general and wide. Thus, three more specific research questions were stated, which address prominent challenges and fingerprint recognition and therefore allow fulfilling the more general research objective:

$\mathcal{RQ}_1$ *"Can DL outperform classical fingerprint sample enhancements?"*

$\mathcal{RQ}_2$ *"Can DL be used for a better orientation field estimation?"*

$\mathcal{RQ}_3$ *"Can DL provide solutions for efficient processing structures?"*

Fingerprint sample enhancement can improve the biometric utility of fingerprint samples significantly. DL-based enhancement even improves the performance on latent fingerprint samples, which can be considered most challenging. CNNs can even be trained directly to improve biometric feature extraction with respect to reliability and accuracy. Thus, DL allows to tailor an enhancement to the environments of data and feature extraction. For some types of data, DL-based approaches already outperform hand-crafted approaches. For other environments, simple hand-crafted approaches work better. Thus, there seems to be no solution yet, which works best for all kind of data. Application of DL shall be therefore considered carefully. But with respect to research question $\mathcal{RQ}_1$ one can conclude from the findings of this thesis and parallel developments, that DL-based approaches are currently superior over hand-crafted approaches especially in the most challenging environments.

The orientation field is a coarse representation of a fingerprint. Estimations of orientation fields need to be reliable and accurate, because they are relevant for many processes in biometric feature extraction and biometric comparison. Results of DL-based approaches at independent benchmark framework FVC-ongoing yield superiority of those approaches over hand-crafted approaches. The currently best approach DEX-OF is state of the art in fingerprint orientation field estimation. It already performs close to reasonable boundaries of performance, which is limited by the accuracy of mark-ups used as ground truth for evaluation. Thus, one can conclude with respect to research question $\mathcal{RQ}_2$, that DL-based approaches currently outperform hand-crafted approaches for fingerprint orientation field estimation.

Three fields of application were evaluated in this thesis: Fingerprint Indexing, alignment, and data origin estimation. The indexing approach developed in this thesis (DIMI) performs well at independent benchmark framework FVC-ongoing. DIMI can be considered state of the art for Fingerprint Indexing besides MCC. DL-based approaches for fingerprint alignment achieved promising results. DL-based approaches can also be used for data origin estimation. But there are no significant differences in performance compared to hand-crafted approaches. The findings with respect to research question $\mathcal{RQ}_3$ allow conclusion, that DL-based approaches can provide improvements to many aspects of efficiency in fingerprint recognition systems.

Finally, one can conclude with respect to the main research objective, that there are many aspects, in which DL already outperforms hand-crafted approaches. Thus, DL can be considered current state of the art for fingerprint orientation field estimation and Fingerprint Indexing. DL was applied to several more aspects, which yield new alternative solutions to hand-crafted approaches. However, DL did not always perform best and in some cases hand-crafted approaches may still be the more appropriate solution.

# Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, volume 16, pages 265–283, 2016.

[2] A. A. Abbood, G. Sulong, and S. Peters. A review of fingerprint image pre-processing. *Jurnal Teknologi*, 69:79–84, 2014.

[3] M. B. Abdallah, J. Malek, A. T. Azar, H. Belmabrouk, J. E. Monreal, and K. Krissian. Adaptive noise-reducing anisotropic diffusion filter. *Neural Computing and Applications*, 27(5):1273–1300, 2016.

[4] S. Abt, C. Busch, and H. Baier. A quality score honoring approach to semantic conformance assessment of minutiae-based feature extractors. In *International Conference of the Special Interest Group on Biometrics (BIOSIG)*, pages 21–32, 2011.

[5] A. Ahmad, I. Arshad, and G. Raja. Partial fingerprint image enhancement using region division technique and morphological transform. *Nucleus*, 52 (2):63–70, 2015.

[6] H. H. Ahmed, H. M. Kelash, M. Tolba, and M. Badwy. Fingerprint image enhancement based on threshold fast discrete curvelet transform (FDCT) and gabor filters. *International Journal of Computer Applications (IJCA)*, 110(3), 2015.

[7] A. H. Ansari. Generation and storage of large synthetic fingerprint database. *ME Thesis, Jul*, 2011.

[8] K. Arora and P. Garg. A quantitative survey of various fingerprint enhancement techniques. *International Journal of Computer Applications (IJCA)*,

28(5):24–28, 2011.

[9] C. Bai, T. Zhao, W. Wang, and M. Wu. An efficient indexing scheme based on k-plet representation for fingerprint database. In *International Conference on Intelligent Computing (ICIC)*, pages 247–257. Springer, 2015.

[10] C. Bai, W. Wang, T. Zhao, and M. Li. Learning compact binary quantization of minutia cylinder code. In *International Conference on Biometrics (ICB)*, pages 1–6. IEEE, 2016.

[11] C. Bai, W. Wang, T. Zhao, and M. Li. Fast exact fingerprint indexing based on compact binary minutia cylinder codes. *Neurocomputing*, 275:1711–1724, 2018.

[12] A. R. Baig, A. R. Baig, and K. Khurshid. Fingerprint enhancement over the past few years. In *Fourth International Conference on Aerospace Science and Engineering (ICASE)*, pages 1–7. IEEE, 2015.

[13] A. R. Baig, I. Huqqani, and K. Khurshid. Enhancement of latent fingerprint images with segmentation perspective. In *11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 132–138. IEEE, 2015.

[14] P. Baldi and Y. Chauvin. Neural networks for fingerprint recognition. *Neural Computation*, 5(3):402–418, 1993.

[15] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

[16] M. V. Bandur, B. M. Popovic, A. M. Raicevic, and D. Randelovic. Improving minutiae extraction in fingerprint images through robust enhancement. In *21st Telecommunications Forum (TELFOR)*, pages 506–509. IEEE, 2013.

[17] J. S. Bartunek, M. Nilsson, B. Sallberg, and I. Claesson. Adaptive fingerprint image enhancement with emphasis on preprocessing of data. *IEEE Transactions on Image Processing*, 22(2):644–656, 2013.

[18] O. Bausinger and E. Tabassi. Fingerprint sample quality metric NFIQ 2.0. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 167–171, 2011.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[20] A. Bazen, R. N. Veldhuis, S. H. Gerez, and M. Sarfraz. Hybrid fingerprint matching using minutiae and shape. *Computer Aided Intelligent Recognition Techniques and Applications*, pages 119–129, 2005.

[21] G. Bebis, T. Deaconu, and M. Georgiopoulos. Fingerprint identification using delaunay triangulation. In *International Conference on Information Intelligence and Systems*, pages 452–459. IEEE, 1999.

[22] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, 2013.

[23] F. Benhammadi, M. Amirouche, H. Hentous, K. B. Beghdad, and M. Aissani. Fingerprint matching from minutiae texture maps. *Pattern Recognition*, 40(1):189–197, 2007.

[24] L. Beslay, J. Galbally, and R. Haraksim. Automatic fingerprint recognition: from children to elderly. Technical report, Joint Research Center, 2018.

[25] B. Bhanu and X. Tan. Fingerprint indexing based on novel features of minutiae triplets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(5):616–622, 2003.

[26] V. G. Biradar and S. H. Article: Fingerprint ridge orientation extraction: A review of state of the art techniques. *International Journal of Computer Applications (IJCA)*, 91(3):8–13, April 2014.

[27] S. Biswas, N. K. Ratha, G. Aggarwal, and J. Connell. Exploring ridge curvature for fingerprint indexing. In *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2008.

[28] S. R. Borra, G. J. Reddy, and E. S. Reddy. An efficient fingerprint enhancement technique using wave atom transform and MCS algorithm. *Procedia Computer Science*, 89:785–793, 2016.

[29] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics (COMPSTAT)*, pages 177–186. Springer, 2010.

[30] A. Bouaziz, A. Draa, and S. Chikhi. A cuckoo search algorithm for fingerprint image contrast enhancement. In *Second World Conference on Complex Systems (WCCS)*, pages 678–685. IEEE, 2014.

[31] A. Bouaziz, A. Draa, and S. Chikhi. Bat algorithm for fingerprint image enhancement. In *12th International Symposium on Programming and Systems (ISPS)*, pages 1–8. IEEE, 2015.

[32] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 3, 2000.

[33] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[34] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 7(04):669–688, 1993.

[35] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese. Fast cartoon + texture image filters. *IEEE Transactions on Image Processing*, 19(8):1978–1986, 2010.

[36] C. Busch, D. Lodrova, E. Tabassi, and W. Krodel. Semantic conformance testing for finger minutiae data. In *Security and Communication Networks (IWSCN), 2009 Proceedings of the 1st International Workshop on*, pages 1–7. IEEE, 2009.

[37] T. Caldwell. Market report: border biometrics. *Biometric Technology Today*, 2015(5):5–11, 2015.

[38] K. Cao and A. Jain. Latent orientation field estimation via convolutional neural network. In *International Conference on Biometrics (ICB)*, pages 349–356, May 2015. doi: 10.1109/ICB.2015.7139060.

[39] K. Cao and A. K. Jain. Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on Information Forensics and Security (TIFS)*, 10(1):104–117, 2015.

[40] K. Cao and A. K. Jain. Automated latent fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[41] K. Cao, J. Liang, and J. Tian. A div-curl regularization model for fingerprint orientation extraction. In *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 231–236. IEEE, 2012.

[42] K. Cao, E. Liu, and A. K. Jain. Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(9):1847–1859, 2014.

[43] K. Cao, E. Liu, and A. K. Jain. Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1847–1859, 2014.

[44] R. Cappelli. Fast and accurate fingerprint indexing based on ridge orientation and frequency. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(6):1511–1521, 2011.

[45] R. Cappelli and M. Ferrara. A fingerprint retrieval system based on level-1 and level-2 features. *Expert Systems with Applications*, 39(12):10465–10478, 2012.

[46] R. Cappelli, D. Maio, and D. Maltoni. SFinGe: an approach to synthetic fingerprint generation. In *International Workshop on Biometric Technologies (BT2004)*, pages 147–154, 2004.

[47] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni. Fingerprint verification competition 2006. *Biometric Technology Today*, 15(7):7–9, 2007.

[48] R. Cappelli, D. Maio, and D. Maltoni. Semi-automatic enhancement of very low quality fingerprints. In *6th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 678–683. IEEE, 2009.

[49] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI*, 32(12): 2128–2141, 2010.

[50] R. Cappelli, D. Maltoni, and F. Turroni. Benchmarking local orientation extraction in fingerprint recognition. In *20th International Conference on Pattern Recognition (ICPR)*, pages 1144–1147. IEEE, 2010.

[51] R. Cappelli, M. Ferrara, and D. Maio. Candidate list reduction based on the analysis of fingerprint indexing scores. *IEEE Transactions on Information Forensics and Security (TIFS)*, 6(3):1160–1164, 2011.

[52] R. Cappelli, M. Ferrara, and D. Maltoni. Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):1051–1057, 2011.

[53] L. Cătălin. Development of optimal filters obtained through convolution methods, used for fingerprint image enhancement and restoration. *Public Administration*, 14(2):20, 2014.

[54] W. Chapman, R. Hicklin, G. Kiebuzinski, P. Komarinski, J. Mayer-Splain, M. Taylor, and R. Wallner. Markup instructions for extended friction ridge features. *NIST Special Publication*, 1151, 2013.

[55] P. Chauhan. Steps in fingerprint enhancement techniques. *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, 4(7), 2014.

[56] F. Chen, X. Huang, and J. Zhou. Hierarchical minutiae matching for fingerprint and palmprint identification. *IEEE Transactions on Image Processing*, 22(12):4964–4971, 2013.

[57] J. Chi and M. Eramian. Enhancement of textural differences based on morphological component analysis. *IEEE Transactions on Image Processing*, 24(9):2671–2684, 2015.

[58] F. Chollet et al. Keras. https://keras.io, 2015.

[59] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20 (3):273–297, 1995.

[60] S. Crihalmeanu, A. Ross, S. Schuckers, and L. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. *Technical Report, WVU, Lane Department of Computer Science and Electrical Engineering*, 2007.

[61] A. Dantcheva, P. Elia, and A. Ross. What else does your biometric data reveal? a survey on soft biometrics. *IEEE Transactions on Information Forensics and Security (TIFS)*, 11(3):441–467, 2016.

[62] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 14(1):21–30, 2004.

[63] J. de Boer, A. M. Bazen, and S. H. Gerez. Indexing fingerprint databases based on multiple features. In *14th Workshop on Circuits, Systems and Signal Processing (CSSP)*. Technology Foundation (STW), 2001.

[64] P. Deshmukh, S. Pathan, and R. Pathan. Image enhancement techniques for fingerprint identification. *International Journal of Scientific and Research Publications (IJSRP)*, 2013.

[65] Digital Persona. FingerJetFX. https://github.com/FingerJetFXOSE/FingerJetFXOSE, 2012.

[66] V. Divya. Adaptive fingerprint image enhancement based on spatial contextual filtering and pre-processing of data. *International Journal of Computing and Technology (IJCAT)*, 1, 2014.

[67] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. Technical report, NIST National Institute of Standards and Technology, 1998.

[68] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue. Fingerprint and on-line signature verification competitions at ICB 2009. In M. Tistarelli and M. Nixon,

editors, *Advances in Biometrics*, volume 5558 of *Lecture Notes in Computer Science*, pages 725–732. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01792-6. doi: 10.1007/978-3-642-01793-3_74. URL http://dx.doi.org/10.1007/978-3-642-01793-3_74.

[69] P. Drozdowski, C. Rathgeb, and C. Busch. Bloom filter-based search structures for indexing and retrieving iris-codes. *IET Biometrics*, 7(3):260–268, 2017.

[70] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12(Jul):2121–2159, 2011.

[71] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[72] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research (JMLR)*, 11(Feb):625–660, 2010.

[73] O. A. Esan, T. Zuva, S. M. Ngwira, and K. Zuva. Performance improvement of authentication of fingerprints using enhancement and matching algorithms. *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, 3(2), 2013.

[74] D. Ezhilmaran and M. Adhiyaman. A review study on fingerprint image enhancement techniques. *International Journal of Computer Science & Engineering Technology (IJCSET) ISSN*, pages 2229–3345, 2014.

[75] M. Fahmy and M. Thabet. A novel scheme for fingerprint enhancement. In *31st National Radio Science Conference (NRSC)*, pages 142–149. IEEE, 2014.

[76] J. Feng and A. Cai. Fingerprint indexing using ridge invariants. In *18th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 433–436. IEEE, 2006.

[77] J. Feng, J. Zhou, and A. K. Jain. Orientation field estimation for latent fingerprint enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(4):925–940, 2013.

[78] G. Li, B. Yang, and C. Busch. A fingerprint indexing algorithm on encrypted domain. In *IEEE Trustcom/BigData/SE/I/SPA*, pages 1030–1037. IEEE, 2016.

[79] A. Gago-Alonso, J. Hernández-Palancar, E. Rodríguez-Reina, and A. Muñoz-Briseño. Indexing and retrieving in fingerprint databases under

structural distortions. *Expert Systems with Applications*, 40(8):2858–2871, 2013.

[80] M. Galar, J. Derrac, D. Peralta, I. Triguero, D. Paternain, C. Lopez-Molina, S. García, J. M. Benítez, M. Pagola, E. Barrenechea, et al. A survey of fingerprint classification Part I: Taxonomies on feature extraction methods and learning models. *Knowledge-based systems*, 81:76–97, 2015.

[81] F. Galton. *Finger prints*. Macmillan and Company, 1892.

[82] M. D. Garris and R. M. McCabe. NIST special database 27: Fingerprint minutiae from latent and matching tenprint images. Technical report, NIST National Institute of Standards and Technology, 2000.

[83] H. Geng, J. Li, J. Zhou, and D. Chen. An improved gabor enhancement method for low-quality fingerprint images. In *Applied Optics and Photonics China (AOPC2015)*, pages 96751J–96751J. International Society for Optics and Photonics, 2015.

[84] D. Gerdeman. India's ambitious national identification program. *Research & Ideas, HBS Working*, 2012.

[85] R. S. Germain, A. Califano, and S. Colville. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, 4(4):42–49, 1997.

[86] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[87] M. Ghafoor, I. A. Taj, W. Ahmad, and N. M. Jafri. Efficient 2-fold contextual filtering approach for fingerprint enhancement. *IET Image Processing*, 8(7):417–425, 2014.

[88] L. Ghiani, V. Mura, P. Tuveri, and G. L. Marcialis. On the interoperability of capture devices in fingerprint presentation attacks detection. In *Italian Conference on CyberSecurity (ITASEC)*, pages 66–75, 2017.

[89] S. Ghosh, S. Roy, U. Kumar, and A. Mallick. Gray level image enhancement using cuckoo search algorithm. In *Advances in Signal Processing and Intelligent Recognition Systems (SIRS)*, pages 275–286. Springer, 2014.

[90] E. Gibney. Google AI algorithm masters ancient game of go. *Nature News*, 529(7587):445, 2016.

[91] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256, 2010.

[92] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[93] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[94] C. Gottschlich. Curved-region-based ridge frequency estimation and curved gabor filters for fingerprint image enhancement. *IEEE Transactions on Image Processing*, 21(4):2220–2227, 2012.

[95] C. Gottschlich, P. Mihailescu, and A. Munk. Robust orientation field estimation and extrapolation using semilocal line sensors. *IEEE Transactions on Information Forensics and Security (TIFS)*, 4(4):802–811, 2009.

[96] C. Gottschlich, T. Hotz, R. Lorenz, S. Bernhardt, M. Hantschel, and A. Munk. Modeling the growth of fingerprints improves matching for adolescents. *IEEE Transactions on Information Forensics and Security (TIFS)*, 6(3):1165–1169, 2011.

[97] S. Greenberg, M. Aladjem, D. Kogan, and I. Dimitrov. Fingerprint image enhancement using filtering techniques. In *15th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 322–325. IEEE, 2000.

[98] P. Grother, M. McCabe, C. Watson, M. Indovina, W. Salamon, P. Flanagan, E. Tabassi, E. Newton, and C. Wilson. Performance and interoperability of the INCITS 378 fingerprint template. Technical report, NIST National Institute of Standards and Technology, 2006.

[99] D. Gunning. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2017.

[100] A. Gyaourova and A. Ross. A novel coding scheme for indexing fingerprint patterns. *Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR)*, pages 755–764, 2008.

[101] A. Gyaourova and A. Ross. A coding scheme for indexing multimodal biometric databases. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 93–98. IEEE, 2009.

[102] A. Gyaourova and A. Ross. Index codes for multibiometric pattern retrieval. *IEEE Transactions on Information Forensics and Security (TIFS)*, 7(2):518–529, 2012.

[103] F. Han, J. Hu, and X. Yu. A biometric encryption approach incorporating fingerprint indexing in key generation. *Lecture Notes in Computer Science*, 4115:342, 2006.

[104] R. M. Haralick, K. Shanmugam, I. Dinstein, et al. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[105] V. Hari, V. J. Raj, and R. Gopikakumari. Unsharp masking using quadratic filter for the enhancement of fingerprints in noisy background. *Pattern Recognition*, 46(12):3198–3207, 2013.

[106] J. Hartloff, J. Dobler, S. Tulyakov, A. Rudra, and V. Govindaraju. Towards fingerprints as strings: Secure indexing for fingerprint matching. In *International Conference on Biometrics (ICB)*, pages 1–6. IEEE, 2013.

[107] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[108] S. He, C. Zhang, and P. Hao. Clustering-based descriptors for fingerprint indexing and fast retrieval. In *Asian Conference on Computer Vision (ACCV)*, pages 354–363. Springer, 2009.

[109] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[110] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(8):777–789, 1998.

[111] T. Hotz. Intrinsic coordinates for fingerprints based on their longitudinal axis. In *6th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 500–504. IEEE, 2009.

[112] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[113] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[114] O. N. Iloanusi. Fusion of finger types for fingerprint indexing using minutiae quadruplets. *Pattern Recognition Letters*, 38:8–14, 2014.

[115] O. N. Iloanusi. Effective statistical-based and dynamic fingerprint preprocessing technique. *IET Biometrics*, 6(1):9–18, 2016.

[116] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[117] A. Iqbal and A. M. Namboodiri. Cascaded filtering for biometric identification using random projections. In *National Conference on Communications (NCC)*, pages 1–5. IEEE, 2011.

[118] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19795-1:2006 Information technology - Biometric Performance Testing and Reporting - Part 1: Principles and Framework. ISO, International Organization for Standardization, 2006.

[119] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19795-2:2007 Information technology - Biometric Performance Testing and Reporting - Part 2: Testing methodologies for technology and scenario evaluation. ISO, International Organization for Standardization, 2007.

[120] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC JTC 1/SC 37 Part 1 Overview Standards Harmonization Document (SC 37 N-SD11). ISO, International Organization for Standardization, Geneva, Switzerland, 2010.

[121] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19795-2:2011 Information technology - Biometric data interchange formats - Part 2: Finger minutiae data. ISO 19795-2:2011, International Organization for Standardization, Geneva, Switzerland, 2011.

[122] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 2382-37:2012 Information technology - Vocabulary - Part 37: Biometrics. ISO 2382-37:2012, International Organization for Standardization, Geneva, Switzerland, 2012.

[123] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 29794-1:2016 Information technology - Biometric sample quality - Part 1: Framework. ISO 29794-1, International Organization for Standardization, Geneva, Switzerland, 2016.

[124] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 29794-4:2016 Information technology - Biometric sample quality - Part 4: Finger image data. ISO 29794-4, International Organization for Standardization, Geneva, Switzerland, 2016.

[125] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 30107-1:2016 Information technology - Biometric presentation attack detection. ISO, International Organization for Standardization, 2016.

[126] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 2382-37:2017(E) Information technology - Vocabulary - Part 37: Biometrics. Technical report, Interna-

tional Organization for Standardization and International Electrotechnical Committee, 2017.

[127] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19794-4:2011 Information technology - Biometric data interchange formats - Part 4: Finger image data. ISO, International Organization for Standardization, 2011.

[128] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19794-8:2011 Information technology - Biometric data interchange formats - Part 8: Part 8: Finger pattern skeletal data. ISO, International Organization for Standardization, 2011.

[129] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 24745:2011 Information technology - Security techniques - Biometric information protection. ISO, International Organization for Standardization, 2011.

[130] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015.

[131] A. Jain and M. Prasad. A novel fingerprint indexing scheme using dynamic clustering. *Journal of Reliable Intelligent Environments*, 2(3):159–171, 2016.

[132] A. Jain and M. V. Prasad. Clustering based fingerprint indexing using triangle spiral. In *11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 81–88. IEEE, 2015.

[133] A. K. Jain and K. Cao. Fingerprint image analysis: role of orientation patch and ridge structure dictionaries. *Geometry Driven Statistics*, 121:288, 2015.

[134] A. K. Jain and S. Minut. Hierarchical kernel fitting for fingerprint classification and alignment. In *16th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 469–473. IEEE, 2002.

[135] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000.

[136] A. K. Jain, S. S. Arora, L. Best-Rowden, K. Cao, P. S. Sudhish, A. Bhatnagar, and Y. Koda. Giving infants an identity: Fingerprint sensing and recognition. In *Eighth International Conference on Information and Communication Technologies and Development (ICTD)*, page 29. ACM, 2016.

[137] M. H. Jakubowski and R. Venkatesan. Randomized radon transforms for biometric authentication via fingerprint hashing. In *ACM workshop on Digital Rights Management*, pages 90–94. ACM, 2007.

[138] U. Jayaraman, A. K. Gupta, and P. Gupta. An efficient minutiae based geometric hashing for fingerprint database. *Neurocomputing*, 137:115–126, 2014.

[139] M. Jazzar and G. Muhammad. Feature selection based verification/identification system using fingerprints and palm print. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, 38(4), 2013.

[140] X. Jia, X. Yang, Y. Zang, N. Zhang, and J. Tian. A cross-device matching fingerprint database from multi-type sensors. In *21st International Conference on Pattern Recognition (ICPR)*, pages 3001–3004. IEEE, 2012.

[141] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *22nd ACM International Conference on Multimedia (ACM-MM)*, pages 675–678. ACM, 2014.

[142] X. Jiang, M. Liu, and A. C. Kot. Fingerprint retrieval for identification. *IEEE Transactions on Information Forensics and Security (TIFS)*, 1(4):532–542, 2006.

[143] W. Kabir. *A new three-stage scheme for fingerprint enhancement and its impact on fingerprint recognition.* PhD thesis, Concordia University, 2013.

[144] W. Kabir, M. O. Ahmad, and M. Swamy. Enhancement of low-quality fingerprint images by a three-stage filtering scheme. In *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1306–1309. IEEE, 2013.

[145] C. Kauba and A. Uhl. Fingerprint recognition under the influence of image sensor ageing. In *4th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2016.

[146] P. Kaur and J. Kaur. A review paper on fingerprint image enhancement with different methods. *International Journal of Modern Engineering Research (IJMER)*, 3(4), 2013.

[147] I. Kavati, M. V. Prasad, and C. Bhagvati. A new indexing method for biometric databases using match scores and decision level fusion. In *Advanced Computing, Networking and Informatics (ICACNI) - Volume 1*, pages 493–500. Springer, 2014.

[148] M. Kayaoglu, B. Topcu, and U. Uludag. Standard fingerprint databases: Manual minutiae labeling and matcher performance analyses. *arXiv preprint arXiv:1305.1443*, 2013.

[149] M. Y. Khachay and M. Pasynkov. Theoretical approach to developing efficient algorithms of fingerprint enhancement. In *International Conference on Analysis of Images, Social Networks and Texts (AIST)*, pages 83–95. Springer, 2015.

[150] M. Y. Khachay, A. Leshko, and A. Dremin. The problem of fingerprint identification: A reference database indexing method based on delaunay triangulation. *Pattern Recognition and Image Analysis*, 24(2):297–303, 2014.

[151] M. A. Khan and T. M. Khan. Fingerprint image enhancement using data driven directional filter bank. *Optik-International Journal for Light and Electron Optics*, 124(23):6063–6068, 2013.

[152] T. M. Khan, M. A. Khan, and Y. Kong. Fingerprint image enhancement using multi-scale ddfb based diffusion filters and modified hong filters. *Optik-International Journal for Light and Electron Optics*, 125(16):4206–4214, 2014.

[153] T. M. Khan, M. A. Khan, Y. Kong, and O. Kittaneh. Stopping criterion for linear anisotropic image diffusion: a fingerprint image enhancement case. *EURASIP Journal on Image and Video Processing (IJIVP)*, 2016(1):1–20, 2016.

[154] T. A. Khandat and I. Shaikh. Improving latent fingerprint matching performance by orientation field estimation using localized dictionaries. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 2014.

[155] J. Khodadoust and A. M. Khodadoust. Fingerprint indexing based on minutiae pairs and convex core point. *Pattern Recognition*, 67:110–126, 2017.

[156] J. Khodadoust and A. M. Khodadoust. Fingerprint indexing based on expanded delaunay triangulation. *Expert Systems with Applications*, 81:251–267, 2017.

[157] D. P. Kingma and J. Ba. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[158] T. Klir. Fingerprint image enhancement with easy to use algorithms. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–4. IEEE, 2015.

[159] M. Kočevar, B. Kotnik, A. Chowdhury, and Z. Kačič. Real-time fingerprint image enhancement with a two-stage algorithm and block–local normalization. *Journal of Real-Time Image Processing (JRTIP)*, pages 1–10, 2014.

[160] K. Komal, D. Albrecht, N. Bhattacharjee, and B. Srinivasan. A region-based alignment-free partial fingerprint matching. In *14th International Conference on Advances in Mobile Computing and Multi Media (MoMM)*, pages 63–70. ACM, 2016.

[161] Z. M. Kovacs-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11):1266–1276, 2000.

[162] R. P. Krish, J. Fierrez, D. Ramos, J. Ortega-Garcia, and J. Bigun. Partial fingerprint registration for forensics using minutiae-generated orientation fields. In *International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2014.

[163] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[164] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems (NIPS*, pages 950–957, 1992.

[165] E. P. Kukula and S. J. Elliott. Implementing ergonomic principles in a biometric system: a look at the human biometric sensor interaction (HBSI). In *40th Annual IEEE International Carnahan Conferences Security Technology*, pages 86–91. IEEE, 2006.

[166] D. G. Kumar, G. Sudha, and B. Revathi. An efficient space partitioning tree approach for indexing and retrieving fingerprint databases. In *International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, page 50. ACM, 2015.

[167] S. Kumar and R. L. Velusamy. Latent fingerprint preprocessing: Orientation field correction using region wise dictionary. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1238–1243. IEEE, 2015.

[168] S. Lawrence, C. L. Giles, and A. C. Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, Institute for Advanced Computer Studies, University of Maryland, 1998.

[169] T. H. Le and T. D. Bui. A codeword-based indexing scheme for fingerprint identification. In *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1352–1356. IEEE, 2008.

[170] T. H. Le et al. A fast and distortion tolerant hashing for fingerprint image authentication. In *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS)*, pages 266–273. Springer, 2009.

[171] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

[172] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[173] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

[174] S.-H. Lee, H. Jeong, and K.-I. Moon. Fingerprint singular point enhancement through angular bandwidth allocation filtering. *International Information Institute (Tokyo). Information*, 18(10):4237, 2015.

[175] S.-O. Lee, Y.-G. Kim, and G.-T. Park. A feature map consisting of orientation and inter-ridge spacing for fingerprint retrieval. In *International Conference on Audio-and Video-Based Biometric Person Authentication (AVBPA)*, pages 184–190. Springer, 2005.

[176] K. Leung and C. H. Leung. Improvement of fingerprint retrieval by a statistical classifier. *IEEE Transactions on Information Forensics and Security (TIFS)*, 6(1):59–69, 2011.

[177] G. Li, C. Busch, and B. Yang. A novel approach used for measuring fingerprint orientation of arch fingerprint. In *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1309–1314. IEEE, 2014.

[178] G. Li, B. Yang, and C. Busch. A fingerprint indexing scheme with robustness against sample translation and rotation. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–8. IEEE, 2015.

[179] G. Li, B. Yang, and C. Busch. A novel fingerprint indexing approach focusing on minutia location and direction. In *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6. IEEE, 2015.

[180] J. Li, W.-Y. Yau, and H. Wang. Fingerprint indexing based on symmetrical measurement. In *18th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 1038–1041. IEEE, 2006.

[181] J. Li, X. Yang, J. Tian, P. Shi, and P. Li. Topological structure-based alignment for fingerprint fuzzy vault. In *19th International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[182] X. Liang, T. Asano, and A. Bishnu. Distorted fingerprint indexing using minutia detail and delaunay triangle. In *3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, pages 217–223. IEEE, 2006.

[183] X. Liang, A. Bishnu, and T. Asano. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *IEEE Transactions on Information Forensics and Security (TIFS)*, 2(4):721–733, 2007.

[184] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[185] E. Liu, J. Liang, L. Pang, M. Xie, and J. Tian. Minutiae and modified biocode fusion for fingerprint-based key generation. *Journal of Network and Computer Applications*, 33(3):221–235, 2010.

[186] F. Liu and D. Zhang. 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model. *Pattern Recognition*, 47 (1):178–193, 2014.

[187] M. Liu, X. Jiang, and A. C. Kot. Fingerprint reference-point detection. *EURASIP Journal on Applied Signal Processing*, 2005:498–509, 2005.

[188] M. Liu, X. Jiang, and A. C. Kot. Fingerprint retrieval by complex filter responses. In *18th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 1042–1042. IEEE, 2006.

[189] M. Liu, X. Jiang, and A. C. Kot. Efficient fingerprint search based on database clustering. *Pattern Recognition*, 40(6):1793–1803, 2007.

[190] M. Liu, X. Chen, and X. Wang. Latent fingerprint enhancement via multiscale patch based sparse representation. *IEEE Transactions on Information Forensics and Security (TIFS)*, 10(1):6–15, 2015.

[191] T. Liu, G. Zhu, C. Zhang, and P. Hao. Fingerprint indexing based on singular point correlation. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages II–293. IEEE, 2005.

[192] T. Liu, C. Zhang, and P. Hao. Fingerprint indexing based on LAS registration. In *IEEE International Conference on Image Processing (ICIP)*, pages 301–304. IEEE, 2006.

[193] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 1, 2017.

[194] D. Lodrova, C. Busch, E. Tabassi, W. Krodel, M. Drahansky, et al. Semantic conformance testing methodology for finger minutiae data. *International Conference of the Special Interest Group on Biometrics and Electronic Signatures (BIOSIG)*, pages 31–42, 2009.

[195] D. G. Lowe. Object recognition from local scale-invariant features. In *Seventh IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157. Ieee, 1999.

[196] A. Lumini, D. Maio, and D. Maltoni. Continuous versus exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, 18(10):1027–1034, 1997.

[197] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

[198] T. Mahashwari and A. Asthana. Image enhancement using fuzzy technique. *International Journal of Research in Engineering Science and Technology (IJRET)*, 2(2):1–4, 2013.

[199] D. Maio and L. Nanni. An efficient fingerprint verification system using integrated gabor filters and parzen window classifier. *Neurocomputing*, 68: 208–216, 2005.

[200] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. FVC2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(3):402–412, 2002.

[201] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2002: Second fingerprint verification competition. In *16th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 811–814. IEEE, 2002.

[202] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. FVC2004: Third fingerprint verification competition. In *Biometric Authentication*, pages 1–7. Springer, 2004.

[203] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer, 2009.

[204] P. Mansukhani, S. Tulyakov, and V. Govindaraju. A framework for efficient fingerprint identification using a minutiae tree. *IEEE Systems Journal*, 4(2): 126–137, 2010.

[205] K. Markert, K. Krehl, C. Gottschlich, and S. F. Huckemann. Detecting anisotropy in fingerprint growth. *arXiv preprint arXiv:1801.06437*, 2018.

[206] Y. Mei, B. Zhao, Y. Zhou, and S. Chen. Orthogonal curved-line gabor filter for fast fingerprint enhancement. *Electronics Letters*, 50(3):175–177, 2014.

[207] J. Merkle, H. Ihmor, U. Korte, M. Niesing, and M. Schwaiger. Performance of the fuzzy vault for multiple fingerprints (extended version). *arXiv preprint arXiv:1008.0807*, 2010.

[208] D. Misra, S. Tripathi, and D. Misra. A review report on fingerprint image enhancement techniques. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 2, 2013.

[209] D. K. Misra, S. Tripathi, and D. Misra. A review report on fingerprint image enhancement filter. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)*, 1(3):403–416, 2013.

[210] S. Mohammedsayeemuddin, S. K. Gonsai, and D. Vandra. Efficient fingerprint image enhancement algorithm based on gabor filter. *International Journal of Research in Engineering and Technology (IJRET)*, 3(4), 2014.

[211] A. Munoz-Briseno, A. Gago-Alonso, and J. Hernandez-Palancar. Fingerprint indexing with bad quality areas. *Expert Systems with Applications*, 40 (5):1839–1846, 2013.

[212] A. Munoz-Briseno, A. Gago-Alonso, and J. Hernandez-Palancar. Using reference point as feature for fingerprint indexing. In *Iberoamerican Congress on Pattern Recognition (CIARP)*, pages 367–374. Springer, 2014.

[213] A. Munoz-Briseno, A. Gago-Alonso, and J. Hernandez-Palancar. Fingerprint matching using a geometric subgraph mining approach. In *Iberoamerican Congress on Pattern Recognition (CIAPR)*, pages 160–167. Springer, 2015.

[214] T. Murakami and K. Takahashi. Fast and accurate biometric identification using score level indexing and fusion. In *International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2011.

[215] A. Nagar, S. Rane, and A. Vetro. Alignment and bit extraction for secure fingerprint biometrics. In *Media Forensics and Security II*, volume 7541, pages 1–14. International Society for Optics and Photonics, 2010.

[216] K. A. Nagati. A non-linear model for fingerprints matching based on minutiae-core pairs interaction. In *22nd International Conference on Computer Theory and Applications (ICCTA)*, pages 174–180. IEEE, 2012.

[217] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[218] L. Nanni and A. Lumini. Descriptors for image-based fingerprint matchers. *Expert Systems with Applications*, 36(10):12414–12422, 2009.

[219] S. Neethu, S. Sreelakshmi, and D. Sankar. Enhancement of fingerprint using FFT×|FFT| n filter. *Procedia Computer Science*, 46:1561–1568, 2015.

[220] D.-L. Nguyen, K. Cao, and A. K. Jain. Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge. In *International Conference on Biometrics (ICB)*, pages 9–16. IEEE, 2018.

[221] R. J. K. Nilam and R. Joshi. Adaptive fingerprint image enhancement for low-quality of images by learning from the images and features extraction. *International Journal of Software and Hardware Research in Engineering (IJSHRE)*, 2, 2014.

[222] NIST. NFIQ2.0: NIST Fingerprint Image Quality 2.0. https://github.com/usnistgov/NFIQ2, 2016.

[223] NIST. NFIQ 2.0 - NIST fingerprint image quality. Technical report, NIST National Institute of Standards and Technology, 2016. URL https://www.nist.gov/document/nfiq2reportpdf.

[224] L. Oehlmann, S. Huckemann, and C. Gottschlich. Performance evaluation of fingerprint orientation field reconstruction methods. In *International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2015.

[225] M. A. Olsen, E. Tabassi, A. Makarov, and C. Busch. Self-organizing maps for fingerprint image quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 138–145. IEEE, 2013.

[226] M. A. Olsen, M. Bockeler, and C. Busch. Predicting dactyloscopic examiner fingerprint image quality assessments. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–12. IEEE, 2015.

[227] M. A. Olsen, V. Šmida, and C. Busch. Finger image quality assessment features–definitions and evaluation. *IET Biometrics*, 2015.

[228] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, et al. MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings-Vision, Image and Signal Processing*, 150(6):395–401, 2003.

[229] J. Ouyang, J. Feng, J. Lu, Z. Guo, and J. Zhou. Fingerprint pose estimation based on faster R-CNN. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 268–276. IEEE, 2017.

[230] N. Pandey and S. Singh. Adaptive latent fingerprint indexing. In *5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)*, pages 509–514. IEEE, 2014.

[231] Parliament and Council of European Union. Parliament and council regulation (EU) no 2018/1861, 2018. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018R1861&from=en.

[232] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems Workshop (NIPS-W)*, 2017.

[233] A. A. Paulino, E. Liu, K. Cao, and A. K. Jain. Latent fingerprint indexing: Fusion of level 1 and level 2 features. In *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[234] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

[235] C. Puri, K. Narang, A. Tiwari, M. Vatsa, and R. Singh. On analysis of rural and urban indian fingerprint images. In *Ethics and Policy of Biometrics*, pages 55–61. Springer, 2010.

[236] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1(2):4, 2017.

[237] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5105–5114, 2017.

[238] R. Rajin and K. Ajith. Comparative study on various fingerprint image enhancement techniques. *Compusoft*, 4(2):1502, 2015.

[239] D. K. Rao, G. Kishore, and G. Vasavi. Adaptive fingerprint enhancement. *International Journal of Future Generation Communication and Networking (IJFGCN)*, 7(4):159–170, 2014.

[240] N. K. Ratha, S. Chen, and A. K. Jain. Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, 1995.

[241] A. Reddy, U. Jayaraman, V. D. Kaushik, and P. Gupta. An efficient fingerprint indexing scheme. In *Second International Conference on Soft Computing for Problem Solving (SocProS)*, pages 723–728. Springer, 2014.

[242] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *International Confenrence on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[243] K. Rerkrai and V. Areekul. A new reference point for fingerprint recognition. In *International Conference on Image Processing (ICIP)*, volume 2, pages 499–502. IEEE, 2000.

[244] A. Ross and A. Jain. Biometric sensor interoperability: A case study in fingerprints. In *International Workshop on Biometric Authentication (IWBA)*, pages 134–145. Springer, 2004.

[245] A. Ross and R. Mukherjee. Augmenting ridge curves with minutiae triplets for fingerprint indexing. In *SPIE Conference on Biometric Technology for Human Identification IV*, volume 6539, page 65390C, 2007.

[246] A. Ross, A. Jain, and J. Reisman. A hybrid fingerprint matcher. *Pattern Recognition*, 36(7):1661–1673, 2003.

[247] R. Rothe, R. Timofte, and L. Van Gool. DEX: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCV-W)*, pages 10–15, 2015.

[248] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge, 2014.

[249] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[250] M. Sahasrabudhe and A. M. Namboodiri. Learning fingerprint orientation fields using continuous restricted boltzmann machines. In *2nd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 351–355. IEEE, 2013.

[251] M. Sahasrabudhe and A. M. Namboodiri. Fingerprint enhancement using unsupervised hierarchical feature learning. In *Indian Conference on*

*Computer Vision Graphics and Image Processing (ICVGIP)*, page 2. ACM, 2014.

[252] M. J. Saks and J. J. Koehler. The coming paradigm shift in forensic identification science. *Science*, 309(5736):892–895, 2005.

[253] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh. On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–7. IEEE, 2014.

[254] H. Sawant and M. Deore. A comprehensive review of image enhancement techniques. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 1(2):39–44, 2010.

[255] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[256] P. Schuch. Survey on features for fingerprint indexing. *IET Biometrics*, June 2018. ISSN 2047-4938. URL http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2017.0279.

[257] P. Schuch, S. Schulz, and C. Busch. De-convolutional auto-encoder for enhancement of fingerprint samples. In *6th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–7. IEEE, 2016.

[258] P. Schuch, S. Schulz, and C. Busch. Minutia-based enhancement of fingerprint samples. In *International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2017.

[259] P. Schuch, S.-D. Schulz, and C. Busch. Convnet regression for fingerprint orientations. In *Scandinavian Conference on Image Analysis*, pages 325–336. Springer, 2017.

[260] P. Schuch, S.-D. Schulz, and C. Busch. Deep expectation for estimation of fingerprint orientation fields. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 185–190. IEEE, 2017.

[261] P. Schuch, S.-D. Schulz, and C. Busch. Intrinsic limitations of fingerprint orientation estimation. *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2017.

[262] P. Schuch, J. M. May, and C. Busch. Estimating the data origin of fingerprint samples. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553235.

[263] P. Schuch, J. M. May, and C. Busch. Learning neighbourhoods for fingerprint indexing. In *2018 14th Signal Image Technology and Internet-based Systems (SITIS)*, pages 1–6, Nov 2018.

[264] P. Schuch, J. M. May, and C. Busch. Unsupervised learning of fingerprint rotations. In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2018. doi: 10.23919/BIOSIG.2018.8553096.

[265] P. Schuch, S. Schulz, and C. Busch. Survey on the impact of fingerprint image enhancement. *IET Biometrics*, 7:102–115(13), March 2018. ISSN 2047-4938. URL http://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2016.0088.

[266] M. Selvi and A. George. FBFET: Fuzzy based fingerprint enhancement technique based on adaptive thresholding. In *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

[267] M. K. Sharma, J. Joseph, and P. Senthilkumaran. Directional edge enhancement using superposed vortex filter. *Optics & Laser Technology*, 57:230–235, 2014.

[268] X. Shuai, C. Zhang, and P. Hao. Fingerprint indexing based on composite set of reduced SIFT features. In *19th International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[269] K. Singh, A. Gupta, and R. Kapoor. Fingerprint image super-resolution via ridge orientation-based clustered coupled sparse dictionaries. *Journal of Electronic Imaging*, 24(4):043015–043015, 2015.

[270] K. Singh, R. Kapoor, and R. Nayar. Fingerprint denoising using ridge orientation based clustered dictionaries. *Neurocomputing*, 167:418–423, 2015.

[271] P. Sood and M. Kaur. Methods of automatic alignment of fingerprint in fuzzy vault: a review. In *Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–4. IEEE, 2014.

[272] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

[273] M. J. Stephen and P. P. Reddy. Design of new parameterized transformation functions and multi objective criterion for fingerprint image enhancement. *Information Science and Technology*, 3(2):54, 2014.

[274] M. J. Stephen, P. Reddy, V. Vasavi, et al. Fingerprint image enhancement through particle swarm optimization. *International Journal of Computer Applications (IJCA)*, 66(21), 2013.

[275] P. Sutthiwichaiporn and V. Areekul. Adaptive boosted spectral filtering for progressive fingerprint enhancement. *Pattern Recognition*, 46(9):2465–2486, 2013.

[276] J. Svoboda, F. Monti, and M. M. Bronstein. Generative convolutional networks for latent fingerprint reconstruction. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 429–436. IEEE, 2017.

[277] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[278] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[279] E. Tabassi and P. Grother. Fingerprint image quality. In *Encyclopedia of Biometrics*, pages 482–490. Springer, 2009.

[280] E. Tabassi, C. Wilson, and C. Watson. NIST fingerprint image quality. *NIST Res. Rep. NISTIR7151*, pages 34–36, 2004.

[281] B. Tams. Absolute fingerprint pre-alignment in minutiae-based cryptosystems. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–12. IEEE, 2013.

[282] Y. Tang, F. Gao, J. Feng, and Y. Liu. Fingernet: An unified deep network for fingerprint minutiae extraction. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 108–116. IEEE, 2017.

[283] S. Tarar and E. Kumar. Fingerprint image enhancement: iterative fast fourier transform algorithm and performance evaluation. *International Journal of Hybrid Information Technology (IJHIT)*, 6(4), 2013.

[284] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL http://arxiv.org/abs/1605.02688.

[285] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[286] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer, 2017.

[287] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528. IEEE, 2011.

[288] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991.

[289] A. M. Turky and M. S. Ahmad. The use of SOM for fingerprint classification. In *International Conference on Information Retrieval & Knowledge Management (CAMP)*, pages 287–290. IEEE, 2010.

[290] F. Turroni, D. Maltoni, R. Cappelli, and D. Maio. Improving fingerprint orientation extraction. *IEEE Transactions on Information Forensics and Security (TIFS)*, 6(3):1002–1013, 2011.

[291] B. T. Ulery, R. A. Hicklin, J. Buscaglia, and M. A. Roberts. Accuracy and reliability of forensic latent fingerprint decisions. *Proceedings of the National Academy of Sciences*, 108(19):7733–7738, 2011.

[292] B. T. Ulery, R. A. Hicklin, J. Buscaglia, and M. Roberts. Repeatability and reproducibility of decisions by latent fingerprint examiners. *PloS one*, 7(3): e32800, 2012.

[293] B. T. Ulery, R. A. Hicklin, M. A. Roberts, and J. Buscaglia. Measuring what latent fingerprint examiners consider sufficient information for individualization determinations. *PloS one*, 9(11):e110179, 2014.

[294] B. T. Ulery, R. A. Hicklin, M. A. Roberts, and J. Buscaglia. Changes in latent fingerprint examiners' markup between analysis and comparison. *Forensic Science International*, 247:54–61, 2015.

[295] B. T. Ulery, R. A. Hicklin, M. Roberts, and J. Buscaglia. Interexaminer variation of minutia markup on latent fingerprints. *Forensic Science International*, 264:89–99, 2016.

[296] D. K. Vandana, D. Singh, P. Raj, M. Swathi, and P. Gupta. Kd-tree based fingerprint identification system. In *2nd International Conference on Anti-counterfeiting, Security and Identification (ASID)*, pages 5–10. IEEE, 2008.

[297] A. Vij and A. Namboodiri. Fingerprint indexing based on local arrangements of minutiae neighborhoods. In *IEEE Computer Society Conference*

*on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 71–76. IEEE, 2012.

[298] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *25th International Conference on Machine Learning (ICML)*, pages 1096–1103. ACM, 2008.

[299] J.-W. Wang, N. T. Le, C.-C. Wang, and J.-S. Lee. Enhanced ridge structure for improving fingerprint image quality based on a wavelet domain. *IEEE Signal Processing Letters (SPL)*, 22(4):390–394, 2015.

[300] X. Wang and M. Liu. Fingerprint enhancement via sparse representation. In *Biometric Recognition*, pages 193–200. Springer, 2013.

[301] Y. Wang, J. Hu, and D. Phillips. A fingerprint orientation model based on 2D fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(4):573–585, 2007.

[302] Y. Wang, P. C. Yuen, and Y.-m. Cheung. Hashing fingerprints for identity de-duplication. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 49–54. IEEE, 2013.

[303] Y. Wang, L. Wang, Y.-M. Cheung, and P. C. Yuen. Learning compact binary codes for hash-based fingerprint indexing. *IEEE Transactions on Information Forensics and Security (TIFS)*, 10(8):1603–1616, 2015.

[304] Z. Wang, S. Chen, C. Busch, and X. Niu. Performance evaluation of fingerprint enhancement algorithms. In *Congress on Image and Signal Processing (CISP)*, volume 3, pages 389–393. IEEE, 2008.

[305] C. Watson, G. Candela, and P. Grother. Comparison of FFT fingerprint filtering methods for neural network classification. Technical report, NIST National Institute of Standards and Technology, 1994.

[306] C. I. Watson. NIST special database 14: Mated fingerprint cards pairs 2 version 2. Technical report, NIST National Institute of Standards and Technology, 2001.

[307] C. I. Watson and C. Wilson. NIST special database 4. *Fingerprint Database, National Institute of Standards and Technology*, 17:77, 1992.

[308] C. I. Watson and C. L. Wilson. Special database 14. *S. Department of Commerce, NIST, Advanced Systems Division, Gaithersburg, Maryland. Citeseer*, 1993.

[309] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. Mccabe, S. Janet, and K. Ko. User's guide to NIST biometric image software (NBIS). Technical report, NIST National Institute of Standards and Technology, 2007.

[310] C. I. Watson, G. P. Fiumara, E. Tabassi, W. J. Salamon, and P. A. Flanagan. Fingerprint vendor technology evaluation. Technical report, NIST, dec 2014. URL http://dx.doi.org/10.6028/NIST.IR.8034.

[311] K. Wei and M. Liu. Latent fingerprint enhancement based on orientation guided sparse representation. In *Chinese Conference on Biometric Recognition (CCBR)*, pages 205–213. Springer, 2016.

[312] H. Xu and R. N. Veldhuis. Binary spectral minutiae representation with multi-sample fusion for fingerprint recognition. In *12th ACM Workshop on Multimedia and Security (SIGMM)*, pages 73–80. ACM, 2010.

[313] J. Xu and J. Hu. Multi-constrained orientation field modeling and its application for fingerprint indexing. In *International Conference on Network and System Security (NSS)*, pages 176–187. Springer, 2015.

[314] B. Yang and C. Busch. Parameterized geometric alignment for minutiae-based fingerprint template protection. In *3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, pages 1–6. IEEE, 2009.

[315] B. Yang, Z. Chen, and C. Busch. Raster image representation of fingerprint minutiae. In *ACM Symposium on Applied Computing*, pages 8–12. ACM, 2011.

[316] J. Yang, J. Shin, B. Min, J. Park, and D. Park. Fingerprint matching using invariant moment fingercode and learning vector quantization neural network. In *International Conference on Computational Intelligence and Security (CIS)*, volume 1, pages 735–738. IEEE, 2006.

[317] J. Yang, N. Xiong, and A. V. Vasilakos. Two-stage enhancement scheme for low-quality fingerprint images by learning from the images. *IEEE Transactions on Human-Machine Systems*, 43(2):235–248, 2013.

[318] J. C. Yang and D. S. Park. A fingerprint verification algorithm using tessellated invariant moment features. *Neurocomputing*, 71(10):1939–1946, 2008.

[319] X. Yang, J. Feng, and J. Zhou. Localized dictionaries based orientation field estimation for latent fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):955–969, 2014.

[320] Z. Yao, J.-M. Le Bars, C. Charrier, and C. Rosenberger. Literature review of fingerprint quality assessment and its evaluation. *IET Biometrics*, 5(3): 243–251, 2016.

[321] S. Yoon, J. Feng, and A. K. Jain. On latent fingerprint enhancement. *Biometric Technology for Human Identification VII*, 7667(1):228, 2010.

[322] S. Yoon, J. Feng, and A. Jain. Latent fingerprint enhancement via robust orientation field estimation. In *International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2011.

[323] S. Yoon, J. Feng, and A. K. Jain. Altered fingerprints: Analysis and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(3):451–464, 2012.

[324] S. Yoon, K. Cao, E. Liu, and A. K. Jain. LFIQ: Latent fingerprint image quality. In *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[325] M. Zahedi and O. R. Ghadi. Combining gabor filter and FFT for fingerprint enhancement based on a regional adaption method and automatic segmentation. *Signal, Image and Video Processing (SIVP)*, 9(2):267–275, 2015.

[326] J. A. M. Zegarra, N. J. Leite, and R. da Silva Torres. Wavelet-based fingerprint image retrieval. *Journal of Computational and Applied Mathematics (JCAM)*, 227(2):294–307, 2009.

[327] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010.

[328] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[329] J. Zhang, R. Lai, and C.-C. J. Kuo. Latent fingerprint segmentation with adaptive total variation model. In *5th IAPR International Conference on Biometrics (ICB)*, pages 189–195. IEEE, 2012.

[330] X. Zhang, Q. Feng, and K. He. A new blind fingerprint alignment algorithm used in biometric encryption. In *International Conference on Computer, Communications and Information Technology (ICCIT)*, 2014.

[331] Q. Zhao and A. K. Jain. Model based separation of overlapping latent fingerprints. *IEEE Transactions on Information Forensics and Security (TIFS)*, 7(3):904–918, 2012.

[332] R. Zheng, C. Zhang, S. He, and P. Hao. A novel composite framework for large-scale fingerprint database indexing and fast retrieval. In *International Conference on Hand-Based Biometrics (ICHB)*, pages 1–6. IEEE, 2011.

[333] W. Zhou, J. Hu, S. Wang, I. Petersen, and M. Bennamoun. Fingerprint indexing based on combination of novel minutiae triplet features. In *International Conference on Network and System Security (NSS)*, pages 377–388. Springer, 2014.

[334] W. Zhou, J. Hu, and S. Wang. Enhanced locality-sensitive hashing for fingerprint forensics over large multi-sensor databases. *IEEE Transactions on Big Data*, 2017.

[335] E. Zhu, J.-P. Yin, G.-M. Zhang, and C.-F. Hu. Fingerprint ridge orientation estimation based on neural network. In *5th WSEAS Int. Conference on Signal Processing, Robotics and Automation (ISPRA)*, pages 158–164, 2006.

[336] K. Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics Gems IV*, pages 474–485. Academic Press Professional, Inc., 1994.

ISBN 978-82-326-4080-5 (printed version)
ISBN 978-82-326-4081-2 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Patrick Schuch

Doctoral Thesis

Doctoral theses at NTNU, 2019:242

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Information Security and Communication
Technology

NTNU

Doctoral theses at NTNU, 2019:242

Patrick Schuch

# Deep Learning for Fingerprint Recognition Systems

NTNU
Norwegian University of
Science and Technology