Master's thesis

2020

Mattis Araya, Eivind Reime

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Mattis Araya
Eivind Reime

Human Pose Estimation Using a Two-Staged Convolutional Neural Network

May 2020

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Human Pose Estimation Using a Two-Staged Convolutional Neural Network

Mattis Araya
Eivind Reime

Computer Science
Submission date: May 2020
Supervisor: Heri Ramampiaro
Co-supervisor: Espen Ihlen
Daniel Groos

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Human Pose Estimation, the task of localizing human joints, has become a popular research field in recent years because of its broad application domain. However, it still remains a challenging task due to occlusions, low resolutions, and overall complexity. We investigate how convolutional neural networks and deep learning techniques can enhance the quality of automated tracking of movements, especially for medical purposes. These techniques can further be employed to track fidgety movements, complex and circular movements of small amplitude, whose absence is a strong indicator of cerebral palsy. An automatization of this tracking process could be of high value, as the qualitative metric of today's methods suffers from the dependency of highly experienced observers and is thus limited in clinical practice.

The vision for this project is to make valuable contributions to the InMotion project, a collaboration between St. Olav's University Hospital and the Norwegian University of Science and Technology. We propose a new, two-staged network architecture in an attempt to improve the prediction quality of extremities. The first stage of the network produces an approximation of all body parts, while the second stage focuses solely on extremities. By exploiting a larger quantity of data and performing high-quality predictions for extremities, our method increases precision for predicted extremities measured at lower thresholds.

Sammendrag

Human pose estimation, metoden for å lokalisere menneskelige kroppsdeler, har i de siste årene blitt et populært forskningsfelt grunnet sitt brede applikasjonsdomene. Til tross for denne popularitet er metoden fortsatt vanskelig å utføre grunnet skjulte kroppsdeler, lavoppløselige bilder og dens generelle kompleksitet. Vi utforsker hvordan konvolusjonelle nevrale nettverk og dyplæringsteknikker kan tas i bruk for å øke kvaliteten på detekteringen av kroppsbevegelser, spesielt for medisinsk bruk. Disse dyplæringsteknikkene kan videre brukes til å detektere fidgety movements, komplekse, sirkulære bevegelser, der fraværet av slike bevegelser er en sterk indikator for cerebral parese. En automatisering av denne detekteringsprosedyren kan være av høy verdi ettersom dagens kvalitative metoder er avhengig av svært erfarne observatører. Metodene har derfor sine begrensninger innenfor medisinsk bruk.

Visjonen til dette prosjektet er å komme med verdifulle bidrag til InMotion prosjektet, et samarbeidsprosjekt mellom St. Olavs Universitetssykehus og Norges teknisk-naturvitenskapelige universitet. Vi foreslår en to-stegs nettverksarkitektur i et forsøk på å forbedre prediksjoner for ekstremiteter. Første steg av nettverket produserer en approksimasjon av alle kroppspunkter, mens andre steg fokuserer på nøyaktig prediksjon av ekstremiteter. Ved å utnytte en betydelig mengde data, og ved å utføre prediksjoner av høy kvalitet for ekstremiteter viser vi til en økning i den totale presisjonen for prediksjon av ekstremiteter.

Preface

This report is written by Mattis Araya and Eivind Reime as a part of the course *TDT4900 - Computer Science, Master's Thesis* at the Norwegian University of Science and Technology. The report is based on previous work conducted in *TDT4501 - Computer Science, Specialization Project*. The project work is conducted as a project of collaboration between St. Olavs University Hospital and the Norwegian University of Science and Technology, called InMotion. While our work is limited to a more narrow part of the project, namely accurate tracking of body parts, we hope that our contribution can improve the overall results for predicting cerebral palsy.

We want to thank our supervisor Heri Ramampiaro for his valuable insights and for giving us the chance to work on this inspiring project. We also want to thank our co-supervisors Espen A. F. Ihlen and Daniel Groos. We are sincerely grateful for your inspiring guidance, patience and generous help throughout this project. It has been a pleasure and an honor to work with you all. A special thanks to Elisabeth Araya for providing us with office spaces during the challenging times of COVID-19.

Table of Contents

Chapter 1

Introduction

1.1 Background and Motivation

In European countries, 6% of all live births are very preterm [1]. The increasing survival rates of children who are born very preterm raise issues about the risks of neurological disabilities and cognitive dysfunction. Cerebral palsy (CP) is a permanent disorder in the development of movement and posture in the developing fetal or infant brain [2] and is one of the major disabilities that result from extremely preterm birth. A study conducted in [1] found that 9% of children born very preterm were diagnosed with CP. Although initial damage cannot be repaired, early identification of CP is essential for initiating treatment while the plasticity of the nervous system is high. Accordingly, this gives a high motivation for accurately diagnosing infants with CP.

Diagnosing CP is a difficult task. The utility is limited by expensive equipment and highly experienced and trained personnel [3]. Based on systematic reviews, the general movement assessment shows the best evidence and strength for accurately predicting CP [4]. This method shows excellent results, with a precision of more than 90%. However, the qualitative metric of this method suffers from the dependency of highly experienced observers and is thus limited in clinical practice [5]. It is also time-consuming, and the outcome is based on subjective opinion.

To overcome these limitations, an automated computer-based method for pattern recognition, independent of experienced observers, would be of high value. With this as motivation, a larger research project was initiated at St. Olav's University Hospital in Trondheim, Norway. Researchers of this project have been working actively over 17 years, collecting video recordings of infant children, exploring the opportunity for an automated system, and assessing the quality of the outcome. One of the benefits of such a system would be that it can be scaled up and used without supervision of trained personnel.

1.2 Problem Statement

During recent years, significant progress in the field of Deep Learning has shown that tasks such as image classification, object detection, and tracking can be done efficiently in real-time. An automated motion analysis system requires to capture body movements accurately, ideally without markers or attached sensors to not affect the movements of infants [6]. Using Convolutional Neural Networks and Deep Learning techniques such as Human Pose Estimation, systems are now able to automatically track the movement of infants with high precision. We further explore this field of Deep Learning by proposing a new pipeline architecture to further increase the accuracy of key point detection in order to track and capture the body movement of infants.

Recent methods for Human Pose Estimation has used multiple datasets to improve the overall precision accuracy of predicted body parts. In this project, we investigate how the usage of more data affects the performance of body tracking.

1.3 Goals and Research Questions

The main goal of this thesis is to explore and implement a new network architecture within the field of Human Pose Estimation for improving the accuracy of predicted body parts. In this thesis, we specifically focus on accuracy on extremities because the accuracy of central body parts is already satisfactory. As part of this, we propose a new pipeline for producing predictions for extremities. More specifically, we can formulate the goals as the following research questions:

RQ 1: *How can the task of Human Pose Estimation be optimized to produce predictions of higher quality for cerebral palsy?*

> **RQ 1.1:** *How can we modify the network architecture to produce higher overall accuracy for predicted body parts measured at lower thresholds?*
>
> **RQ 1.2:** *How can we increase key point accuracy of the model merely based on exploiting available data?*

1.4 Outline

In Chapter 2, we introduce relevant background information within both the medical and the technical field. We start by defining cerebral palsy and techniques used to diagnose it. We further provide a brief introduction to Artificial Intelligence and Computer Vision and subsequently give a thorough explanation of Human Pose Estimation. Chapter 3 gives a summary of today's state-of-the-art methods within Human Pose Estimation, as well as other methods related to our work. Chapter 4 describes our methodology and the proposed method for producing predictions of higher quality for extremities. Chapter 5 documents the results produced during the research, and compares our proposed method to other existing state-of-the-art methods. Chapter 6 evaluates both the results and the applicability of our proposed method. Finally, Chapter 7 presents the conclusion for this thesis and suggestions for future work.

Chapter 2

Background

This chapter contains an introduction to important medical and technical theory used as a basis for our research. The content in this chapter is based on our work conducted in *TDT4501 - Computer Science, Specialization Project* which is a preface of the master thesis itself.

2.1 Medical Background

In the following section we take a brief look at the medical background that forms the fundamental motivation for our thesis. We start by defining cerebral palsy and its challenges, before we go on to describe which methods are used to predict and diagnose cerebral palsy in today's society.

2.1.1 Cerebral Palsy

Cerebral palsy (CP) is a well-recognized neurodevelopment condition developed in early childhood and persisting throughout the lifespan. Rosenbaum [2] defines CP as follows:

"Cerebral palsy (CP) describes a group of permanent disorders of the development of movement and posture, causing activity limitation, that is attributed to nonprogressive disturbances that occurred in the developing fetal or infant brain. The motor disorders of cerebral palsy are often accompanied by disturbances of sensation, perception, cognition, communication, and behavior, by epilepsy, and by secondary musculoskeletal problems."

The human brain is complex, and each child diagnosed with CP will have a different outcome and forecast. With this as motivation, the gross motor function [7] was developed in 1997. This method classifies children with CP into five levels of mobility based on the key function of severity [8, 9]:

- **GMFCS Level I: Walks without Limitations**
  Children and youth perform gross motor skills such as climbing and running, but more complex skills such as coordination and balance are limited.
- **GMFCS Level 2: Walks with Limitations**
  Children and youth are capable of walking, but may find it difficult to walk long distances and needs railings or other supporting devices in most settings to climb stairs.
- **GMFCS Level 3: Walks Using a Hand-Held Mobility Device**
  Children and youth require hand-held mobile devices such as canes or crutches in order to walk outside, and wheeled mobility for long-distance walks.
- **GMFCS Level 4: Self-Mobility with Limitations; May Use Powered Mobility**
  Children and youth use powered mobility such as an electric wheelchair. The person actively controls a joystick for maneuvering.
- **GMFCS Level 5: Transported in a Manual Wheelchair**
  Children and youth require physical assistance in all settings. Their ability is also limited in order to maintain in trunk postures.

2.1.2 Fidgety Movements

Detection of children with a developmental disorder, specifically CP, is both a challenging and tedious process. The diversity reflects the difficulties in techniques used in the field of medicine to assess the brain at an early stage. These techniques range from clinical observations, requiring no technical equipment, to more sophisticated methods such as ultrasound and magnetic resonance imaging.

In recent years, a new method for neuromotor assessment of infants has been developed. This method is based on the assessment of general movements. General movements are movements of the fetus and young infant in which all parts of the body participate [10]. General movements that typically occur at 3-5 months post-term are defined as fidgety movements and are usually the predominant movement pattern for awake infants in this time period [11]. Prechtl [12] defined the movements as *circular movements of small amplitude, moderate speed, and variable acceleration of neck, trunk, and limbs in all directions*. The movements are complex, occur frequently, and last long enough to be observed correctly. Figure 2.2 shows two infants, where the leftmost panel displays an infant born at term. This infant presents fidgety movements, as can be seen from the continuous change in position. Respectively, the rightmost panel shows an infant born at week 28. This infant displays abnormal general movements, which can be interpreted from the lack of variation in movements. The absence of fidgety movements poses a strong indication for later neurological impairments, especially for CP [13]. Figure 2.1 shows the strong predictive value and correlation between the absence of fidgety movement and cerebral palsy. A systematic review was also conducted on 326 children in 2013 and showed a sensitivity of 98% and a specificity of 91% by utilizing the absence of fidgety movements [4]. The sensitivity measures the proportion of infants with cerebral palsy where the condition is correctly identified and specificity measures the percentage of healthy infants correctly identified as healthy.

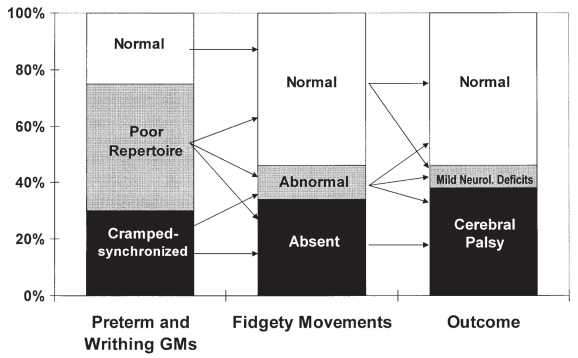

Figure 2.1: A longitudinal study of 130 infants with its respective ultrasound findings. From left, preterm, and writhing quality preceding the quality of fidgety movements, which corresponds to the neurological outcome at three years (right column) [14].

2.1.3 Assessment Procedure

The evaluation of general movements and their complexity is demanding and requires highly trained personnel. Gestalt perception is a well-known method for evaluating the movements of infants [15]. The method is a powerful, yet vulnerable instrument in the analysis of complex phenomena [16]. In order to provide a reliable assessment of recorded general movements of infants using gestalt perception, a standardized framework has been developed [17]. The infant is recorded in spine position, with neutral clothes and preferably with bare arms and legs. Active wakefulness is the ideal state of the infant for preserving the best quality of assessment. An example of these standardized recordings can be viewed in Figure 2.2. It is important for the observer to focus on the overall movement and not pay attention to details. This is because environmental distractions may interfere with the observer's gestalt perception.

Despite gestalt perceptions robustness, the method has some limitations and can be prone to error. An observer's assessment of general movements is subjective, which may lead to different outcomes based on the selected observer. The method also demands experienced observers in order to obtain a reliable diagnosis. For less-experienced personnel, it can, for example, be difficult to distinguish between abnormal general movements and seizures. This is because general movements with low range can show successive movement components that are similar to stereotyped movements of subtle seizures [14].

2.2 Technical Background

In this section, we describe the fundamental techniques in the field of Artificial Intelligence, followed by an introduction to Computer Vision theory that is both related to this project and important to understand in order to grasp the aspects of Human Pose Estimation. We further give a brief introduction to more specific techniques in the domain of Computer Vision, which is highly relevant to this project, namely data augmentation, evaluation metrics and backbone networks.

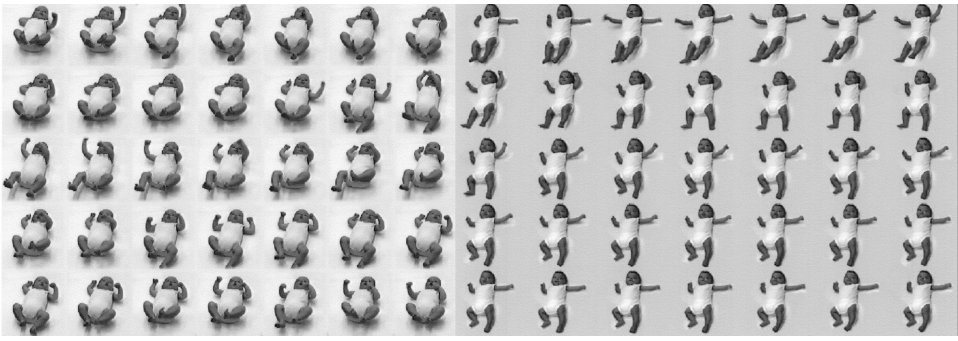

Figure 2.2: Recordings of fidgety movements on infant children [10].

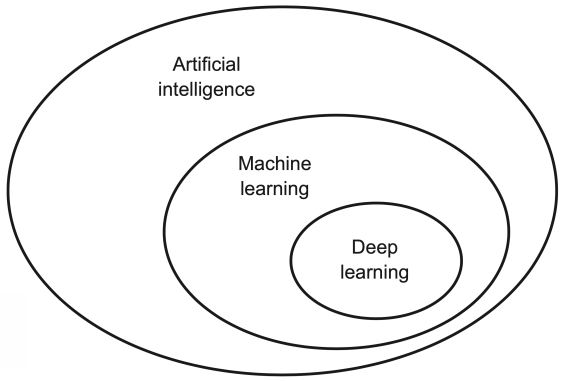

Figure 2.3: Artificial intelligence, machine learning and deep learning [19].

2.2.1 Artificial Intelligence

Since the break of dawn, humans have tried to understand the fundamentals of this world. One of the most interesting questions is: how do we think? The field of Artificial Intelligence was created based on this question in 1956 [18], when a group of pioneers wanted to explore whether computers could be able to think like humans. In [19], the following definition is stated: *AI is the effort to automate intellectual tasks normally performed by humans*.

The field of AI encompasses Machine Learning and Deep Learning (see Figure 2.3), but it also includes other areas that do not include learning. In the beginning, experts thought that human-level Artificial Intelligence could be achieved by defining an explicit set of rules for the computer system as a base for its knowledge manipulation. Today, this field of AI is known as *symbolic AI*. The method was suitable to solve a well-defined problem, such as playing chess, but struggled with more complex problems such as translation of natural language and image segmentation. Thus, it created the need for a new area in AI, *Machine Learning*.

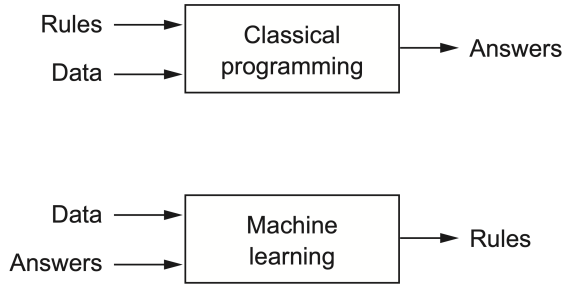

Figure 2.4: Symbolic AI VS machine learning [19].

2.2.2 Machine Learning

In symbolic AI, the programmers creates a set of rules for the system as well as feeding it with input data. Thus, the system is only capable of doing and learning what its creator specifies. Machine Learning proposes a new way of thinking; a system should be able to learn how to perform a specific task without human intervention. Thus, it should learn patterns and draw conclusions on its own.

By providing a Machine Learning system with data and the expected answers, it is capable of learning patterns and build a set of rules, all by itself. These rules can be reapplied to new data to retrieve the potentially unknown answers. We say that the Machine Learning system is trained, rather than being explicitly programmed (see Figure 2.4).

Machine Learning tasks are classified into several categories, whereas the two most commonly known are supervised and unsupervised learning. Supervised learning describes systems provided with both the input data and the corresponding answers, producing a mathematical model able to predict the answers of new data. Unsupervised learning is building a mathematical model solely based on the input data, which is further used to uncover patterns in data or grouping input into categories.

2.2.3 Deep Learning

Deep Learning (DL) is a subfield of Machine Learning trying to learn representations from data through successive layers, whereas each layer focuses on a distinct set of features based on output from the previous layer. The word deep in Deep Learning does not describe how much information the network extracts from its data, but rather how many hidden layers that make up the network. According to various experts, a deep neural network is a neural network consisting of at least three layers, thus at least one hidden layer. Due to the number of layers, a deep neural network is capable of learning patterns of data with millions of properties, all without human intervention.

Artificial Neural Network

An artificial neural network (ANN) is a computing system based on the biological neural networks found in the brain. The system consists of interconnected processors, called neurons, producing a specific output based on its input. We say that the neuron is activated

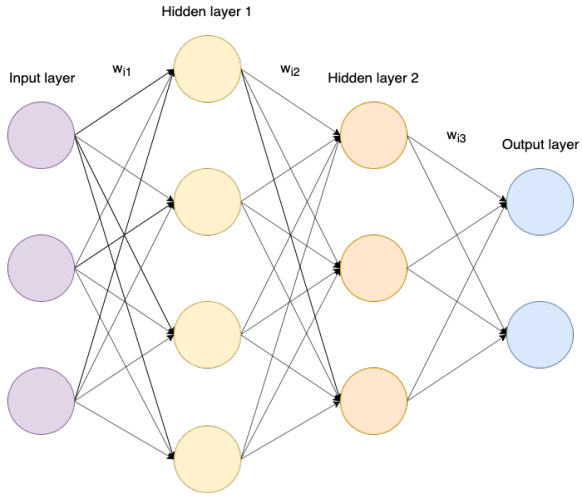

Figure 2.5: A fully connected network with two hidden layers.

when it produces an output. Synapses make up the connections between neurons, making them able to ”communicate” with each other. A neural network consists of an input layer and an output layer with a collection of hidden layers in between them (see Figure 2.5). Each layer consists of several nodes or neurons, each given a specific weight. The importance of the information floating through the network is decided by reviewing the weight of each layer.

To be able to determine how the network performs, we need to specify an evaluation function, the loss function. The loss function takes the output, or the prediction, from the network, compares it with the expected output and calculates a distance score. An optimizer takes this distance score to adjust the value of the weights in such a way that further reduces the distance score. Initially, the weights of the network are set to an arbitrary value, but by repeating this training loop, we can adjust the weight, little by little, to minimize the loss function (see Figure 2.6). Eventually, the predictions of the network are as close as they can be to the target, and we say that the network is trained.

Convolutional Neural Network

Convolutional Neural Networks, or CNNs, are networks that perform a linear mathematical operation called convolution. This particular type of network is primarily used to process data with a known grid-like topology [20]. The most significant difference between CNNs and other densely connected networks is that a dense layer learns global patterns in their input data, while a convolution layer learns local patterns. Thus, a convolutional layer can learn a pattern at a specific spot in an image and recognize the same pattern at a different location, without having to learn it again like the dense layer. This property makes CNNs data efficient when it comes to processing images.

Take, for example, a CNN trained to classify images. An image is often represented as an array of pixel values, and the first layer in such a network would typically extract

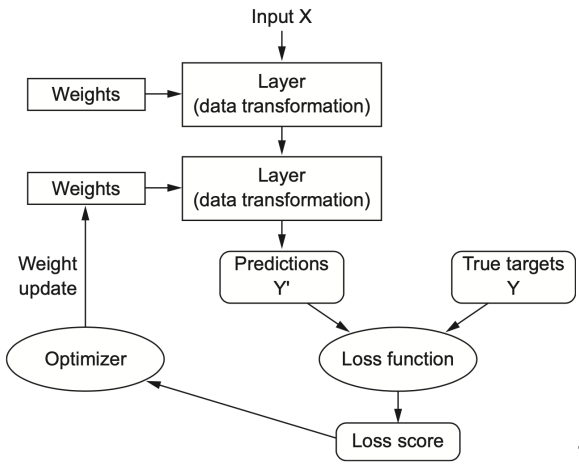

Figure 2.6: Overview of a neural network [19].

information about the presence or absence of edges in the image. The second layer typically extracts information about a specific collection of edges, regardless of position in the image. Successive layers might look at objects made out of these groups of edges. Thus, each successive layer extracts more and more complex features from the image by combining the features learned in previous layers (see Figure 2.7).

A convolutional layer takes two arguments as primary input: a *feature map*, usually a multidimensional array of data, and a *kernel* or *filter*. The filter can be seen as a field of view in the layer and is often a lot smaller in spatial size than the input. During convolution, the filter moves across the feature map focusing on extracting information about a specific set of features. A layer has a set of filters that makes up the *depth* of the layer. Every convolutional layer produces an output called the *output feature map* (see Figure 2.8), which is used as input for the next layer.

Evaluation Metrics

In order to evaluate how well specific algorithms models the given data, several evaluation metrics, referred to as *loss functions*, has been developed. These functions reveal the difference between the estimated values and ground truth values and measures the quantity of data that will be minimized during training. As neural networks take as many shortcuts as possible, it is crucial to select the right loss function according to the problem being solved. Fortunately, for common problems such as regression and classification, there have been conducted much research, creating guidelines for choosing the correct loss function. For regression problems, where one is trying to predict continuous values, *Mean square Error (MSE)* is one of the most common loss functions. This function measures the loss by calculating the square sum of the difference between the predicted value and the ground truth value, over all data points, divided by the number of data points, as shown in Equation 2.1. As derived, the method penalizes more substantial errors more harshly than smaller errors. The output results are always positive, regardless of the predicted value. Large positive

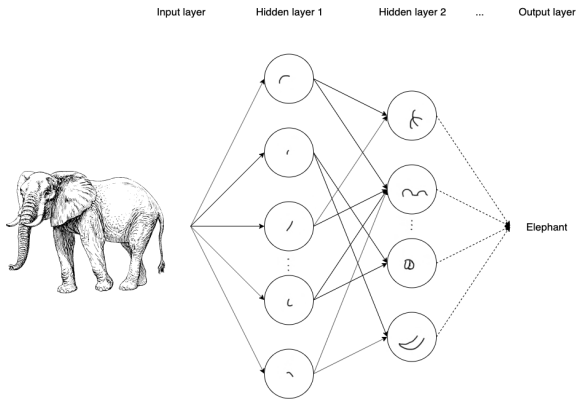

Figure 2.7: The spatial hierarchy of visual modules, used by the neural network to classify the input image as an elephant.

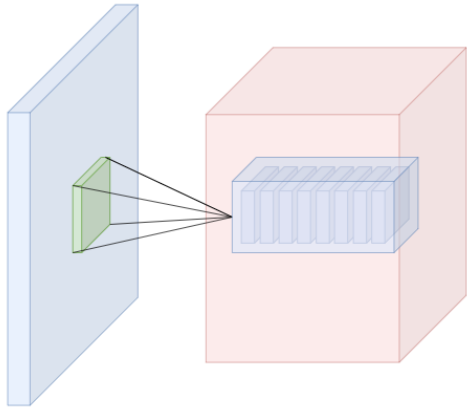

Figure 2.8: Illustration of a convolution layer with a depth of eight, thus the layer contains eight filters. The green box represent the view port of a filter.

values indicate a greater distance between the predicted value and the ground truth value. Hence a perfect output value is 0.0.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y_j})^2 \tag{2.1}$$

Another function used for real-valued regression tasks is the *Euclidean Loss* function. This method takes in the back-propagated value x and calculates how far this input is from the expected targets t using Equation 2.2. This error function is not parameterized by any weights w. As MSE, this method also penalizes larger errors.

$$EuclideanLoss = \frac{1}{2}\sum_{i=1}^{m}(x_i - t_i)^2 \tag{2.2}$$

Because classification problems are trying to solve a different task, where predicted values are categorized from a set of finite pre-defined values, other measurements for loss is required. *Cross-Entropy Loss* is one of the most common functions used for two-class classification problems, where the output value increases as the predicted probability diverge from its ground truth label. As derived from Equation 2.3, we see that the penalty score is logarithmic and will provide low scores for small differences between the predicted value $\hat{y_i}$ and the true value y_i, while substantial differences will produce higher scores.

$$CrossEntropyLoss = -(y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})) \tag{2.3}$$

For many-class classification problems, *Categorical Cross-Entropy Loss* is most commonly used. This function is a combination of a *Softmax Activation* and a Cross-Entropy Loss function. The main difference compared to standard cross-entropy loss, is that because only one result can be correct, the true class is represented as a one-hot encoded vector. Hence, the loss is measured by calculating how close the predicted value $\hat{y_i}$ is to the vector, as shown in Equation 2.4.

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * \log(\hat{y_{ij}})) \tag{2.4}$$

2.2.4 Data Augmentation

In systems of deep neural networks, *overfitting* is a recurring problem. Overfitting happens when the network model specializes in the training data set and does not generalize well for new data. One way to avoid overfitting is by feeding the network with even more training data. The problem is: in many situations, this extra set of data is not available. Data augmentation is a technique for generating more data from an existing dataset, significantly improving performance in tasks like image classification and object detection [21, 22]. By making minor alternations to existing data, we can generate new and unique data that contribute to generalize a model even further (see Figure 2.9). There exist many augmentation techniques, and the most popular ones are the following:

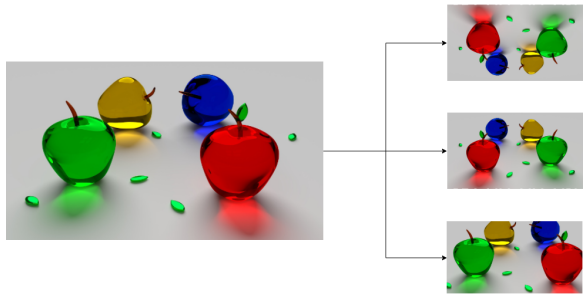

Figure 2.9: Illustration of data augmentation on an image, using rotating, flipping and scaling.

- **Flip:** An image can be flipped horizontally, vertically, or both. It is important to think about whether flipping in a specific direction is useful in the task at hand. If the task is about analyzing cars on the road, vertically flipping an image would not make any sense. Who drives their car upside down?
- **Rotation:** When rotating an image, one might be changing the dimensions in the image. Rotating a square image by 90 degrees would preserve image size while rotating by 60 degrees would not. The issue of preserving image dimensions can be avoided by employing other techniques like padding or cutting.
- **Crop:** Cropping takes a random section from the original image to create a new one. Resizing an image back to original size after cropping is a well-known method called random cropping
- **Scale:** An image can be scaled inward or outward, respectively increasing or decreasing the image size.
- **Translation:** Involves moving an image along the width, height, or both. This method is especially useful for CNNs, because it forces the network to look for an object or pattern in all sections of an image.

2.2.5 Backbone Networks

The research area of Machine Learning is vast, and therefore it is important to review previous and related work before starting on a new task. In cases of neural networks, new problems can often be solved by using already known networks as a baseline to avoid duplicate and unnecessary work. Backbone networks are the baseline networks on which people base their research. OpenPose [23], a real-time multi-person 2D pose estimation network, is an example of this. The network uses another network VGG [24] as a backbone network to initialize the analysis of an image. EfficientNet [25] is an example of a popular backbone network in CNNs, developed as a mobile-size network. The team behind EfficientNet developed a family of models, *EfficientNets*, optimizing both accuracy and floating-point operations per second (FLOPS) by scaling network width, depth, and resolution uniformly.

A technique tightly connected to the usage of backbone networks is *Transfer Learning*. A large dataset is often needed to achieve satisfying results in deep learning, but training a model from scratch on these datasets will be both costly and time-consuming. Transfer learning is the process of reusing an already trained model to a different but related problem. We can employ the technique in three ways:

1. If a new model fully reuse a model and its weights, we call it a *pre-trained model*. This might be useful in scenarios where the original problem is closely related to the new problem, and both datasets are quite similar. An example could be reusing a model trained on a dataset only containing adults on the problem of infant pose estimation.

2. A new model can use part of a pre-trained model as a baseline to extract generic features before doing further processing. In this case, the pre-trained model is known as a *feature extractor* and its weights remain fixed throughout the whole learning process.

3. As in 2, we use a pre-trained model as a baseline, but instead of fixing the weights, we train them together with the rest of the model.

In this project, we use EfficientNet as our backbone network. We employ this backbone with technique 3, where the network is pre-trained on ImageNet [26]. As described above, we further train the pre-trained weights together with the rest of the model.

2.3 Human Pose Estimation

As defined by Leonid Sigal, Human Pose Estimation (HPE) is the task of estimating the configuration of the human body from an image [27]. This also includes the search for a specific body pose, which, in essence, is a set of connected coordinates used to describe the pose of a person. In a simple case, as shown in Figure 2.10, a single-person algorithm can be performed to locate the human limbs, such as the left or right shoulder, neck, and the top of the head. Because of its diverse abundance of applications that can profit from this technology, it is considered as one of the most important problems of Computer Vision. Despite being assessed and researched for many years, it is still considered as a difficult task to solve. The difficulties are many, but the most common and challenging problems are as follows:

- Variance of human visual appearance in an image
- Different light conditions
- The complexity of the human skeletal structure
- Small and barely visible joints

As one of our main goal of this project is to increase the accuracy of detected keypoints, the field of Human Pose Estimation is highly relevant and attractive for this thesis.

Figure 2.10: A human skeleton consisting of 17 keypoints, representing limbs of the human body.

2.3.1 Datasets

The remarkable progress in the field of deep learning and computer vision is much due to the leveraging of large-scale image datasets. A satisfactory amount of labeled data is crucial for both accurate training of models for Human Pose Estimation and to prevent over-fitting. Because of HPE's diverse application domain, there has been developed several open-source datasets for testing, training, and evaluation. One of the most commonly used datasets is the MPII Human Pose dataset [28]. This is a state-of-the-art benchmark for the evaluation of HPE. The images show single or multiple persons performing every day human activities scraped from YouTube videos. Each image is labeled with an activity label, and the dataset contains a total of 410 different human activity labels. Another dataset widely used in Computer Vision and HPE is the COCO dataset [29]. This dataset consists of images of everyday scenes containing common objects in their natural context. The COCO dataset displays more complex everyday scenes compared to the MPII dataset. This is because the goal of the COCO dataset is also to question object recognition in the context of a wider question, scene understanding. Accordingly, this dataset has a wide range of applications and was developed to address the following three core research-problems: detecting non-ionic views of objectives, contextual reasoning between objects, and 2D localization of objects. An example of the extensive labeling for an image can be viewed in Figure 2.11. One last dataset, developed in 2017, is the HSSK dataset [30]. This dataset was developed for three specific tasks, namely human key point detection, caption detection for the Chinese language, and attribute-based zero-shot recognition, which contains both visual and semantic attributes to the objective. Only images labeling human key points will be relevant for this project. The dataset also contains a visibility flag for each annotated key point. This visibility flag, v_i can have three different values, where $v_i = 1$ means the key point is labeled, $v_i = 2$ indicates that the key point is labeled but not visible, and finally $v_i = 3$ indicates that the key point is not labeled. Figure 2.12 shows an example picture taken from the HSSK dataset, displaying bounding boxes and human key points for two humans. As shown, the different key points are connected as segments and not as a fully connected human skeleton as in the MPII dataset. For example, one

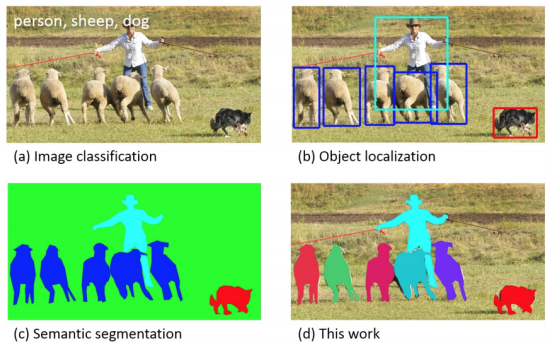

Figure 2.11: The COCO dataset introduces a large, richly-annotated dataset and can be used for image classification, object localization, and semantic segmentation [29].

Datasets	Images	Humans	Keypoints
MSCOCO	200K	250K	17
MPII	25K	40K	16
HKD (HSSK)	270K	511K	14

Table 2.1: Comparison of human keypoint datasets.

can see that the left shoulder, left elbow, and left wrist is connected, forming an individual segment, but the left shoulder is not connected to the neck.

For comparison purposes, Table 2.1 shows the corresponding scope of each dataset. We observe that the HSSK and COCO datasets contain a significantly larger amount of data compared to the MPII dataset.

2.3.2 Approaches

To solve the problem of Human Pose Estimation, various solutions have been proposed. The utilization of Deep Learning-based methods to extract tolerable features from meta-data has produced excellent results, outperforming non-deep state-of-the-art methods. The problem of HPE can first be classified into two categories, namely single-person pose estimation and multi-person pose estimation. While single-person approaches, essentially want to solve a regression problem where the number of keypoints is implicitly stated, multi-person approaches need to solve an unconstrained problem. This is because of the number of positions and humans is unknown.

Single-Person Approaches

The single-person problem in HPE is the most straightforward problem to solve as the human keypoints are implicitly stated, given the fact that there is only one person in the picture. There are two common approaches for the single-person pipeline: direct regression-based framework and heatmap-based framework [31]. As the title suggests,

Figure 2.12: An example picture taken from the HSSK dataset showing bounding boxes and annotated human key points for two humans [30].

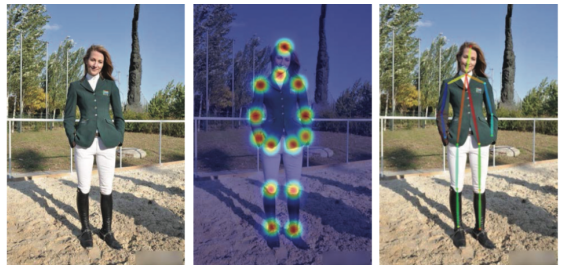

Figure 2.13: Heatmap-based framework for a single person, where (a) shows the original image, (b) illustates the generated heatmaps, and (c) indicates the predicted result [31].

direct regression-based frameworks use regression to predict human keypoints directly. However, studies on pose estimation have shown that his method is highly non-linear because it is challenging to learn mapping directly from feature maps without other procedures [32]. Another drawback of this method is that it can not be applied to solve multi-person problems. Because of this disadvantage, most solutions are developed using a heat map-based framework. This method first regresses heatmaps in order to locate the keypoints, as illustrated in Figure 2.13. The heatmaps are then further used to create the predicted joints.

Multi-Person Approaches

Finding body parts for multi-person problems is a considerably more difficult task than single-person problems. First, neither the position nor the number of people in a picture is given for a multi-person problem. Second, the making of associations between body

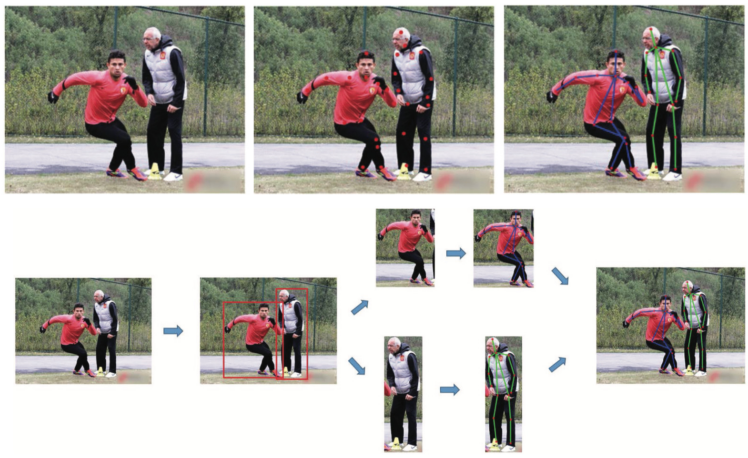

Figure 2.14: Visual comparison of top-down pipeline versus bottom-up pipeline [31].

parts is more difficult due to contact and interactions between people causing occluded joints. Third, the runtime complexity grows with the number of people in the image. Based on these difficulties, two pipelines have been proposed: (1) top-down pipeline and (2) bottom-up pipeline.

- *Top-down approach*: The top-down approach starts with the detection of all humans in a given picture, where each human is segmented into a bounding box. The method then crops the picture based on the resulting bounding boxes and performs keypoint detection for each cropped picture. The resulting picture will accordingly contain the human skeleton with keypoints for each human in the input image. A visualization of the top-down pipeline can be viewed in Figure 2.14, as the approach showed at the bottom of the figure.

- *Bottom-up approach*: The bottom-up approach, is, in essence, a reversed process of the top-down approach. The method first detects keypoints for each human in the image, which is a second stage that is assembled and associated with human instances. A visualization of the bottom-up pipeline can be viewed in Figure 2.14, as the approach showed at the top of the figure.

Both of these pipeline frameworks have been explored using Deep Learning methods in recent years. However, there is no correct answer to which method one should prefer. This is because multiple aspects should be considered in real-world applications, most importantly, speed and accuracy. Accuracy can objectively be measured by the results on some of the keypoint challenges hosted on the most known datasets. Both winners of the HSSK Challenge and the COCO dataset challenge in 2017 employed the top-down pipeline. However, when it comes to speed, the top-down pipeline needs to estimate the pose of each person one by one, which gives a linear run time and increases linearly with the number of humans. In comparison, the images in the bottom-up approach only need to pass through the network once.

2.3.3 Common Evaluation Metrics

Some standardized metrics has been defined in order to objectively measure the performance of Human Pose Estimation models.

Percentage of Correct Parts - PCP

PCP [33] is a standard evaluation metric used on many benchmarks. This metric measures the detection rate of limbs. A limb is evaluated as correctly detected if the distance between the two predicted joint locations and the true limb locations is less than half of the total limb length, respectively denoted as PCP@0.5. Intuitively, a high PCP means a high detected percentage and an accordingly accurate model. This metric has some drawbacks as it penalizes shorter limbs more than other limbs because the shorter limbs, such as lower arms, produce lower thresholds as it is harder to detect.

Percentage of Correct Keypoints - PCK

PCK measures the total percentage of correctly detected keypoints. A keypoint is considered correctly detected if the distance between the true joint and the predicted joint is within a certain threshold. This threshold is commonly 0.5, denoted as PCKh@0.5, which means it considers all predicted keypoints placed within 50% of the head bone link as correctly detected. This method addresses the penalization problem of shorter limbs in PCP since shorter limbs have smaller head bone links. Respectively, a higher PCKh score means a higher percentage of correctly placed keypoints, thus a more accurate model. In this thesis, we will try to increase the PCKh@0.1 of the developed model in the InMotion project.

Percentage of Detected Joints - PDJ

PDJ metric measures accuracy according to the torso. A joint is correctly detected if the distance between the predicted joint and the ground truth location is within a given fraction of the torso, which can vary from the definition. The torso diameter is defined as the distance between the left shoulder and right hip [34]. This means that all joint accuracies are measured with the same error threshold.

Object Keypoint Similarity - OKS

The OKS measures the similarity between the predicted joints and the ground truth joints in a different manner. The main idea is, in essence, to calculate the weighted euclidean distance between the predicted keypoints and the ground truth keypoints. The OKS for a human figure *p* is given by the following formula:

$$OKS_p = \frac{\sum_i \exp\{\frac{-d_{pi}^2}{2s_p^2\sigma_i^2}\}\delta(v_{pi} = 1)}{\sum_i \delta(v_{pi} = 1)} \tag{2.5}$$

where:

- d_i is the Euclidean distances between each ground truth and detected keypoint.
- v_i is the visibility flags of the ground truth.
- s_p is the scale factor for a human figure p.
- k_i is the constant for each keypoint

2.3.4 Summary

As described in the previous sections, there are many factors to consider in order to choose an approach for solving the task of Human Pose Estimation. As the goal of this project is to improve the accuracy of extremities for medical usage, we limit our scope to single-person approaches. We further use a heatmap-based framework because direct regression-based frameworks are highly non-linear. We use both the MPII dataset and the HSSK dataset for training and testing purposes, where the MPII dataset is chosen for its extensive usage, and the HSSK dataset is chosen because methods performing state-of-the-art results have used a combination of MPII and HSSK. Lastly, we use PCKh as a metric throughout the thesis to measure the performance of our model.

Chapter 3

State-of-the-art

As seen in the last chapter, there are many approaches for solving the problem of Human Pose Estimation. In the following chapter, we describe some of the methods which have produced state-of-the-art results. We then take a look at two methods that have similarities to our proposed method. Parts of this chapter are based on our work conducted in *TDT4501 - Computer Science, Specialization Project* which is a preface of the master thesis itself.

3.1 Human Pose Estimation

Human Pose Estimation serves as a fundamental tool for solving many high-level problems such as tracking, human-computer interaction, and human action recognition. Despite the rapid development in HPE, it still remains a challenging problem. Low resolutions, occlusions, and complex variances of body poses are some of the most common challenges in the field. However, new methods have enabled the development of smart implementations in order to deal with these problems. In the following section, we take a look at three methods which have produced state-of-the-art results in the field of HPE, both regarding accuracy and speed.

3.1.1 OpenPose

OpenPose [23] is a state-of-the-art, open-source model for multi-person 2D pose estimation in real-time. While the main focus of many HPE-methods has been on finding body parts of individuals, OpenPose presents an efficient way of finding body parts for multiple persons, performing with competitive results on multiple public benchmarks. Using Part Affinity Fields (PAFs), the method presents the first bottom-up approach for finding a representation of the association between body parts (see Figure 3.1). PAFs are a set of 2-dimensional vectors that encodes the location and orientation of limbs over the image domain. Another way of finding the association between body parts is by detecting an additional midpoint between each pair of parts on a limb. This, however, has its limitations

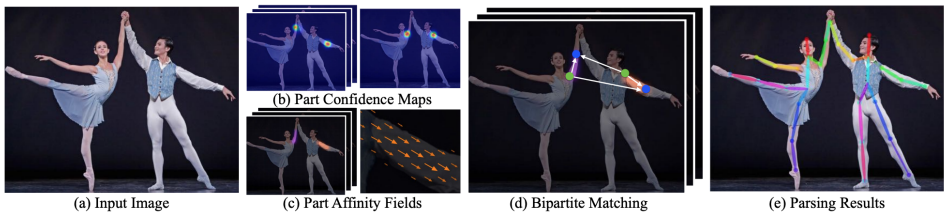

Figure 3.1: OpenPose pipeline, where the complete image is used as input for a CNN to (b) predict confidence maps and (c) part affinity fields. (d) Bipartite matching is further used to produce body-part candidates. (e) The figure shows the final results with fully assembled body poses [23].

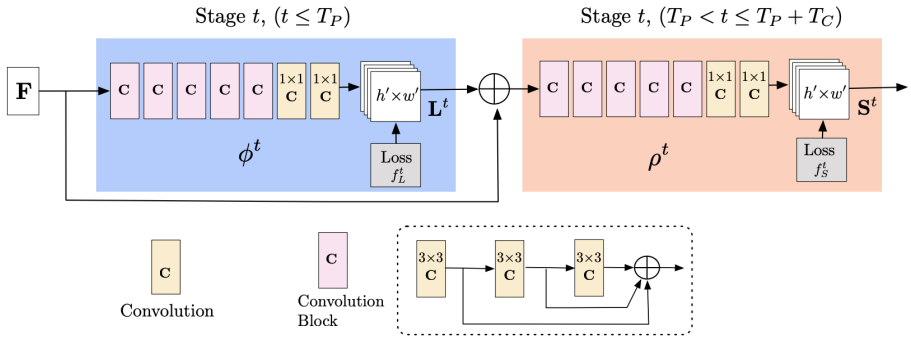

Figure 3.2: OpenPose architecture showing the multi-stage CNN.

as these midpoints can produce false associations between body parts as people crowd together. These false associations are a result of the limitation in representation because the midpoints only encode the position, not the orientation of each limb. Part Affinity Fields solves this problem by how the 2D-vectors are represented: they encode the direction of points from one part of the limb to the other part.

The model consists of 3 consecutive 3x3 kernels, shown in Figure 3.2, which gives a total of only 51 computational operations. The first stage ϕ^t, predicts Part Affinity fields for each limb while the consecutive stage p^t produces confidence maps for each key point. A loss function is applied at the end of each stage in order to iteratively guide the network to predict more accurate PAFs and confidence maps.

Using a three scale search, OpenPose produces a state-of-the-art performance of 75.6% mAP, which indicates the effectiveness of Part Affinity fields to associate body parts. The most remarkable result here is OpenPose's result of only 0.005 seconds to process an image. This demonstrates how a greedy parsing algorithm can produce high-quality results for body parses while preserving runtime performance.

3.1.2 Cascade Feature Aggregation

Cascade Feature Aggregation (CFA) [35] is one of the most recent methods that cascades several hourglass networks to form a robust and efficient model for Human Pose Esti-

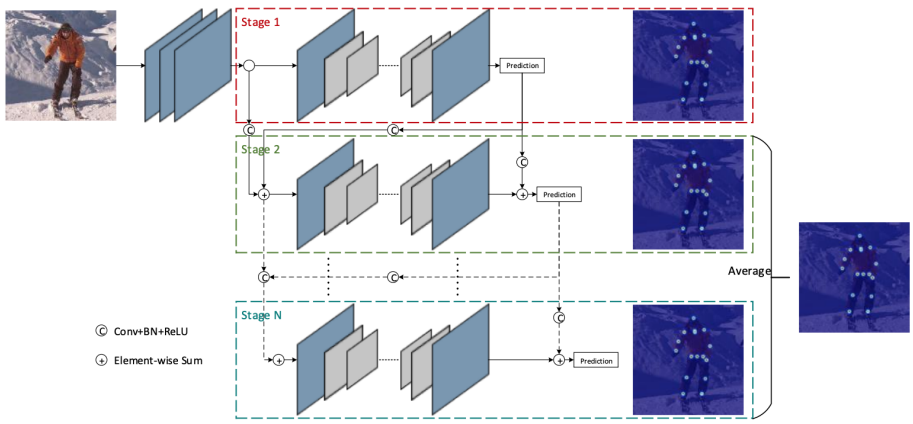

Figure 3.3: Cascade Feature Aggregation architecture showing stages 1-N, where each stage produces new key point predictions based on the inputs and outputs of the previous stage [35].

mation. By aggregating features in different stages, the model obtains a large amount of contextual information. This gives a model that produces accurate body poses while maintaining robustness regarding partial occlusions and low resolution. The resulting work outperforms state-of-the-art methods and achieves the best performance of 93.9% on the MPII benchmark.

The hourglass network has produced promising results and is generally perceived as a sound basis architecture for Human Pose Estimation. Stacked Hourglass [36] produced a PCKh@0.5 of 90.9% on the MPII benchmark by stacking several of these hourglass networks to achieve a robust architecture. Furthermore, multiple attempts have been made in order to improve the backbone network for each stage of the stacked hourglass method. Ke et al. [37] proposed a model, improving the hourglass model with four extensions: (1) *multi scale supervision* for improvement on contextual features, (2) *multi scale regression network* at the end of the network to improve structural matching of multi-scale features, (3) *structure-aware loss* to increase the matching of key points, and (4) a *key point masking training* scheme which makes the network more robust for localizing occluded key points. The method scored a PCKh@0.5 of 92.1% on the MPII benchmark. Li et al. [38] further improved the PCKH@0.5 by adding cross-stage feature aggregation and coarse-to-fine supervision, obtaining a PCKh@0.5 of 92.6%. The model also produced the best performance in the COCO keypoint challenge 2018. The main difference between the original model proposed from Newell [36], and the CFA is that Stacked Hourglass only take the outputs from previous stages as input for the current stage, meanwhile, CFA uses both the inputs and the outputs from the previous stage (see Figure 3.3) as inputs for the next stage. This improves the PCKh@0.5 to a staggering 93.9% on the MPII benchmark.

In order to perform state-of-the-art results, CFA is trained on both the MPII and the HSSK dataset. The model performs a PCKh@0.5 of 89.95% while only being trained on the MPII dataset and PCKh@0.5 of 92.15% with training on both datasets with quad-stage. This demonstrates how crucial additional data is for improving performance. The model

performs an overall best performance while also achieving the best performance on each of the evaluated joints. Results are compared with both a triple-stage model and a five-stage model. The model produces the highest performance with a five-stage model. This is because the results of the first stage may fail on images where people interact, and two bodies intersect, which leads to partially occluded body parts. The last stage (5th) adopts global semantic features and achieves nearly perfect results for the problem of partially occluded body parts.

3.1.3 Toward Fast and Accurate Human Pose Estimation via Soft-gated Skip Connections

Bulat et al. [39] propose a new method that combines the Hourglass [36] and U-Net architectures [40] into a hybrid network which increases performance without increasing the number of parameters due to a smaller number of identity connections within the network. The main focus of this paper is to achieve high accuracy without using computationally heavy neural networks, a research area of HPE, which has received little attention so far.

Residual connections have proven to be extremely important in deep neural networks, and are used by all current state-of-the-art methods. Despite this, the authors of [39] argue that these connections may hinder models from achieving the highest accuracy possible. Hence, they introduce *soft-gated residual connections* defined as:

$$x_{l+1} = \alpha x_l + F(x_1, W_l),$$

where $x_l \in \mathbb{R}C \times w \times h$ are the input features from the previous layer, W_l is a set of weights associated with the lth residual block and F is a residual function implemented using a set of convolutional layers. This soft-gate parameter is used to filter out redundant information in the residual module in such a way that only the useful information is adapted from the previous stage.

The hybrid network structure minimizes the number of identity connections within the network, which increases the overall performance without increasing the number of parameters. Instead of adding the features from two distinct distributions in the residual module, the network concatenates features and combines them using a set grouped convolutional layers, as shown in Figure 3.4, one group for each data source.

As a result, this model achieves state-of-the-art results, surpassing all previous results on the MPII dataset both in terms of accuracy and run-time performance.

3.2 Related Work

In the following section, we describe two methods that are related to ours because they implement cascaded architectural pipelines. Even though these models present results that are far from today's state-of-the-art performance, the methods have some of the same baseline goals as our model.

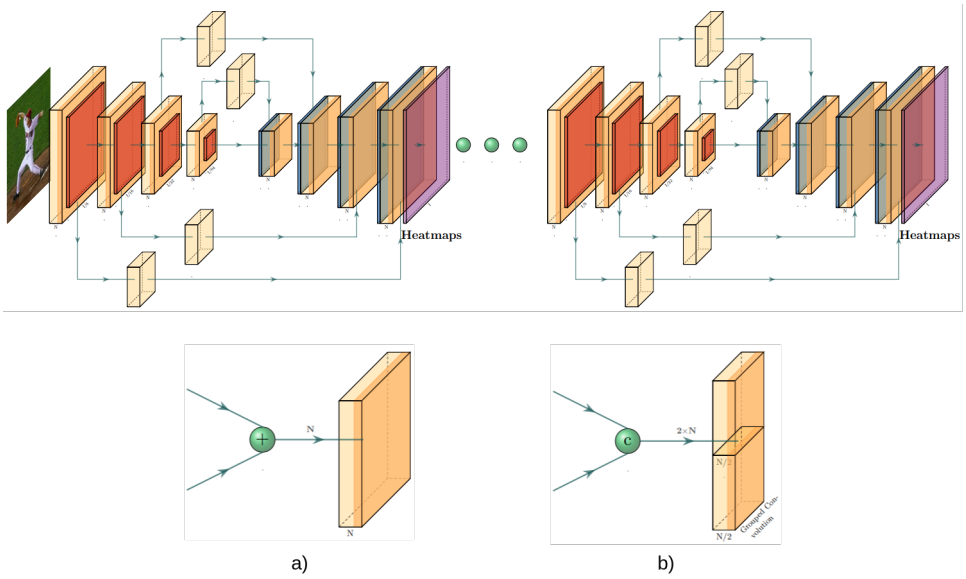

Figure 3.4: Overall network architecture of the proposed method in Bulat et al. [39] showing two ways for aggregating features from the skip connections. a) shows the baseline method [36], merging features using element-wise summation, and b) shows the proposed method in which features are concatenated and then processed using a grouped convolutional layer with a kernel of size 3×3.

3.2.1 Joint Training of a Convolutional Network

Thompson et al. [41] propose a new hybrid architecture consisting of a Deep Convolutional Network and a Markow random field (MRF) [42]. The network consists of a ConvNet Part-Detector 3.5 (state-of-the-art when the paper was published) and a part-based spatial model, which together make up a unified learning framework. The part detector takes an RGB image containing one or more humans as input and gives a key-point heatmap as output. By incorporating a multi-resolution input with overlapping receptive fields, the network is able to see a more significant portion of the input image without affecting the number of weights to a greater extent. An advantage of the Sliding-Window model in Figure 3.5 is the translation-invariant detector, but the cost of the model evaluation is a significant drawback, due to the redundant convolutions in the network.

The part-detector itself predicts a heatmap containing several false-positives and poses that are anatomically incorrect. Therefore, the paper proposes a spatial-model to constraint the connection between joints and enforce consistency in the global pose. The model connects every body part to itself and other body parts to create a fully connected graph. By using convolutional priors, the pair-wise potentials are calculated and used as a basis to remove the false-positives.

The model in this paper combines the part-detector and the spatial-model to a single unified model. During training, they firstly train the part-detector and compute and store the heatmaps separately. Secondly, the spatial model is trained with these heatmaps. Finally, they back-propagate through the entire network. The model is trained and tested on

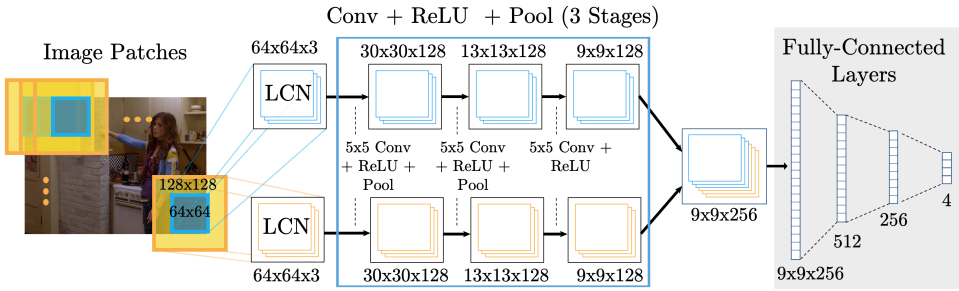

Figure 3.5: Illustration of the multi-resolution sliding-window model with overlapping receptive fields, as proposed in [41].

the FLIC dataset, outperforming all existing architectures within the field of Human Pose Estimation in 2014.

3.2.2 Efficient Object Localization

Thompson et al. [43] propose a method for recovering the spatial accuracy lost as a result of pooling and sub-sampling layers. The architecture is somewhat similar to the architecture pipeline presented in this thesis, and the model is inspired by the multi-resolution ConvNet architecture presented in Thompson et al. [41]. Efficient Object Localization uses an additional convolutional network to utilize the localization results of coarse heat-maps. Figure 3.6 shows the cascaded architecture consisting of a heat-map-based parts model and an additional model used for fine-tuning. In comparison to other cascaded architectures, CFA reuse computed convolutional features in order to reduce the number of parameters, as well as using this as a regulator for the coarse heat-map model. The Coarse heat-map model is, as it implies, responsible for the rough localization. This model returns coarse (x,y) coordinates that are used to crop the convolutional features for each joint. The additional ConvNet uses these inputs to fine-tune heat-maps, providing more accurate predictions for each joint. This is illustrated in Figure 3.3, where we see that the refinements (Δx, Δy) are used with the results from the coarse heat-map model to produce the final predictions.

Compared to the proposed architecture in this thesis, the efficient object localization only uses one additional network to fine-tune the heat-maps for each joint. This is implemented as a Siamese network [44], where the number of instances corresponds to the number of predicted joints. Figure 3.7 shows the siamese network for 14 instances, where each instance forms a convolutional sub-network with four layers. All sub-networks are connected to a 1x1 Convolution at the end that outputs a heat-map. Both the biases and weights of each module are replicated across all instances and are also updated together during backpropagation. The features do, however, not share the same spatial context since the location of each joint is different. As a consequence, the model can perform redundant computations if two cropped windows overlap. However, the researchers of this method have found that this is rare in practice. Since this is a rather ”historic” paper as a result of the rapid progress in Human Pose Estimation, the presented method scores a PCKh@0.5

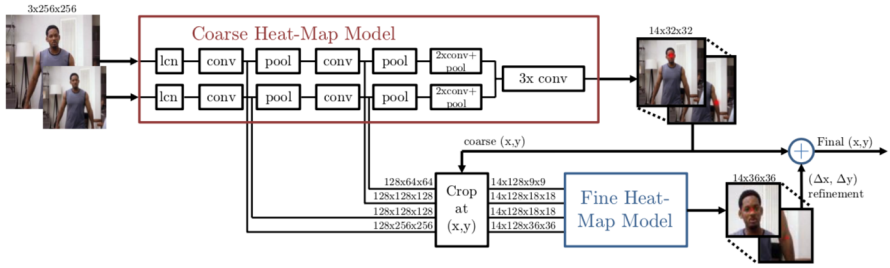

Figure 3.6: Overall pipeline showing the cascaded architecture [43].

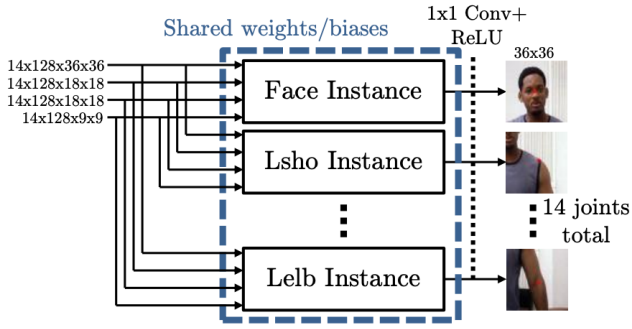

Figure 3.7: Illustration of the fine-heat map model with the siamese network architecture [43].

of only 82.0% on the MPII benchmark. Note that this was state-of-the-art results at the time.

3.2.3 Other Methods

In addition to Thompson et al. [41, 43] research on network models with cascaded architecture, there exist other methods that have tried exploiting the concept of decomposing the human pose into smaller sub-problems in order to overcome the challenges of articulated Human Pose Estimation. Felzenszwalb et al. [45] first introduced deformable part models that benefit from spatial models for the localization of each part of the human body. Many algorithms have later tried to improve the DPM-based architecture [46, 47, 48]. Johnson et al. [49] introduced models using a cascaded architecture for body part detection. Although these methods showed acceptable performance at the time of their publishing, they are, compared to today's state-of-the-art methods, outdated and outperformed. Another common characteristic between these approaches is that they all use some form of handcrafted features, which is now known for poor generalization performance.

Chapter 4

Method

In the upcoming chapter we introduce a method to improve the accuracy of predicted body parts at a lower threshold, respectively at a PCKh measured at 10%. What makes this method unique is not the network implementation itself, but merely the way we modify the overall architectural structure by adding small, efficient networks for the extremities, in an attempt to produce a more accurate final confidence map.

4.1 Background

In the upcoming section we describe the motivation behind our method, how our network models is built, how they are unique, and lastly, how they serve as good baseline models for our purpose.

4.1.1 Motivation

The evaluation metric that has become the standard in HPE is PCKh measured at 50%. As described in Section 2.3.3, this means that all predicted key points placed within 50% of the head bone link is considered as correctly detected. As this has become the standard benchmark, it seems that the goal of most models developed these days is to beat this score. While a PCKh@0.5 is considered sufficient in more generalized systems, this precision does not always suffice in clinical usage. Imagine an automated clinical system that should detect fidgety movements, as defined in Section 2.1.2, by observing and analyzing the limbs of the human body. A threshold of 50% within the head bone link would not be considered as a sufficient accuracy for analyzing the limbs, as body parts like elbows and wrists could be placed quite wrong (see Figure 4.1). While the best benchmark for PCKh@0.5 is at 94.1%, the best-published results for stricter thresholds, such as a PCKh@0.1 is only at about 36%. Also, most articles only present PCKh results for thresholds of 50% and higher as this seems to be the motivation for improvement. Since the key point predictions in the InMotion project are further used to predict CP, a higher

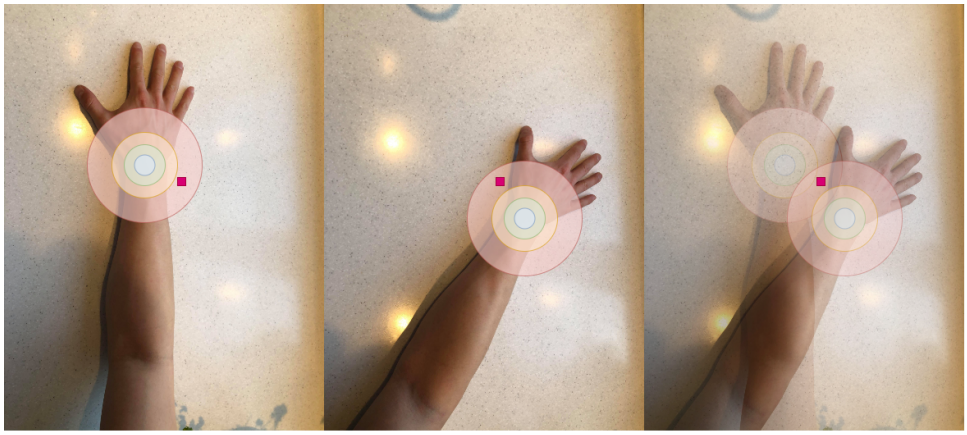

Figure 4.1: An example of a worst-case prediction for a wrist in a sequence of two frames captured from a video. The transparent circles show the PCKh thresholds of 100%, 50%, 30%, 10%, respectively. We can see that the prediction in both a) and b) is well within the PCKh@1 threshold. Based on these predictions, a neural network would conclude with no movement between the two frames, as shown in c), while the wrist actually moved significantly.

precision within a stricter threshold seems most beneficial. This forms the motivation for our proposed method.

4.1.2 Baseline Models

In this section, we introduce the two baseline models of which our pipeline is built upon, namely *Stacked hourglass* and *EfficientNet*. We further explain why these networks are advantageous and why they serve as good baseline models for our purpose.

Staked Hourglass Networks

The *Stacked hourglass network* [36] was developed in 2016, and was built upon the general need to capture information at every scale. At a high level, the model consists of multiple hourglass-shaped modules, and seems very similar to fully convolutional networks. The module differs from the design of fully convolutional networks in its symmetric distribution between pooling and upsampling, which produces an hourglass-like architecture as shown in Figure 4.2. The main idea for the symmetric distribution is that different aspects of understanding the body pose, like arrangement of their limbs and the person's orientation, are best recognized at different scales. While the method produces results that are somewhat lower than the state-of-the art models, it serves as a novel and intuitive architecture that can capture all features across scales.

The method applies convolutional and max pooling layers in order to process features down to a low resolution. This is known as bottom-up processing, where features go from higher to lower resolutions. As seen from Figure 4.2, the network branches off at each

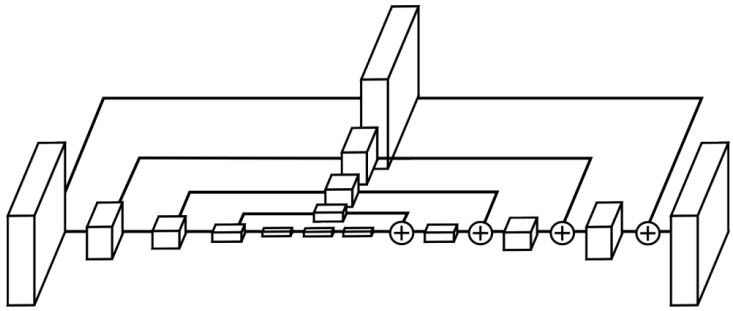

Figure 4.2: Illustration of the hourglass architecture, Newell et al. [36].

max pooling layer, and applies convolutions to the the block at the top of the figure which is the original block with the original resolution. The network proceeds by upsampling and combining features when the lowest resolution is reached. As in Tompson et al. [41], nearest neighbor upsamling of the lower resolution combined with addition of the two sets of features is done in order to bring the information together. Finally, two 1x1 convolutions are executed in order to produce the final confidence maps which has the same resolution as the input.

The hourglass architecture serves as a cheap and novel architecture structure, while still producing sufficient accuracy for key point prediction. This in an important feature for the main network in our approach. The network gives the same resolution for both input and output. This feature certainly comes in handy in our proposed method for the smaller networks, where it is ideal to have a network that can take low input resolution without dramatically decreasing the output resolution.

EfficientNet

The family of *EfficientNets* was introduced in 2019 as a new way to scale models based on available resources [25]. By using a simple but highly efficient compound coefficient, the team is able to propose a method to scale all the dimensions (depth, width, and resolution) uniformly. See Figure 4.3 for a summary of model scaling. Scaling techniques have been used widely in the area of ConvNets before [50, 51, 52], but they all focus on arbitrarily scaling of the dimensions, a process that demands lots of manual tuning without any guarantee of a performance boost. The scaling dimensions are dependent in the following way: higher resolution images should be supplemented with a deeper and wider network. Thus, the paper proposes a compound scaling method using a compound coefficient ϕ to scale the dimensions mentioned above:

$$\begin{aligned} \text{depth: } d &= \alpha^{\phi} \\ \text{width: } w &= \beta^{\phi} \\ \text{resolution: } r &= \gamma^{\phi} \end{aligned}$$

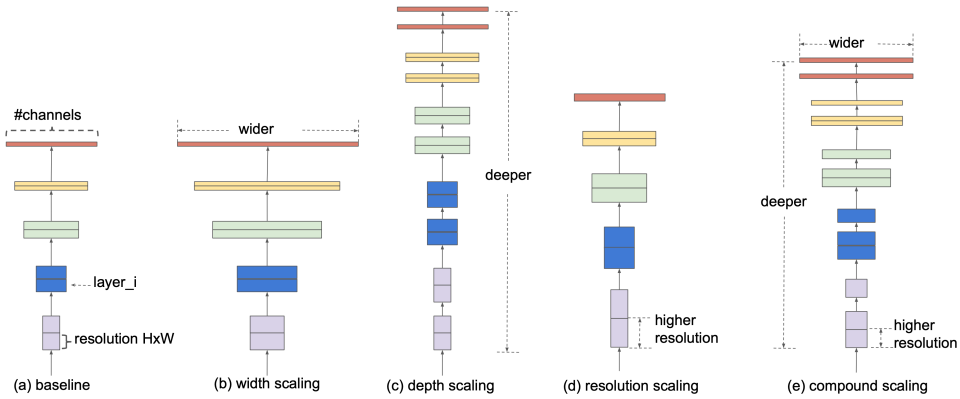

Figure 4.3: Illustration of the different model scaling techniques [25], where (a) is a baseline network example, (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution, and (e) is the compound scaling method that uniformly scales all three dimensions with a fixed ratio.

such that $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$. Thus, a doubling of the network depth will double the FLOPS, whereas doubling resolution or width will increase the FLOPS by a factor of four. Compound scaling starts with the baseline model EfficientNet B0, the least heavy model, and is done by firstly fixing $\phi = 1$ and do a small grid search for α, β, γ. Secondly, fix α, β, γ as constants and scale up the baseline network with varying values of ϕ. By doing this, we can scale the baseline network EfficientNet B0 up to EfficientNet B1 to B7.

The EfficientNet architecture consists of one main building block: the mobile inverted bottleneck MBConv with squeeze-and-excitation optimization (see Figure 4.4). MBConv comes in two versions, one with six times upscaling and one without upscaling at all. Both versions consist of three features [53]:

1. **Depthwise separable convolution**: Splits a standard convolution into two separate layers; depthwise convolution and pointwise convolution. The block employes this technique to reduce computational cost with a minimum loss of accuracy.

2. **Linear bottlenecks**: Assuming that the manifold of interest in a neural network is set in a low-dimensional subspace, the manifold can be spotted by using linear bottlenecks in the convolutional layers. This technique is used to avoid too much information loss from non-linearities.

3. **Inverted residuals**: Appears similar to a standard residual block, but uses shortcuts between the bottlenecks to improve the ability of a gradient to propagate across multiple layers. This results in a considerably more memory-efficient approach.

By employing this family of networks as a baseline for our proposed method, we get a high performing backbone network, both in regards to maximizing accuracy and minimizing FLOPS, that scales up very efficiently based on the input size to the network.

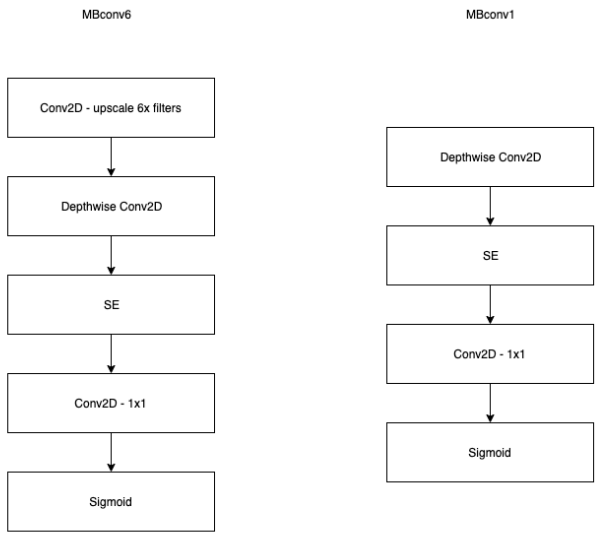

Figure 4.4: The two different MBconv-blocks used in EfficientNet.

4.2 Architecture

The pipeline architecture of our proposed method can be divided into two parts: A, B, as shown in Figure 4.5. Part A consists of an hourglass shaped EfficientPose B4 network with 18 M parameters. The purpose of this network is to produce coarse confidence maps for all key points. These confidence maps are further upsampled and used as input in part B of the architecture. This part consists of multiple EfficientPose B1 networks, which produce new confidence maps for each extremity. Finally, these confidence maps are mapped together with the initial coarse confidence map in order to produce the final confidence map containing all key point predictions. As seen from Figure 4.5, the pipeline takes images of size 1024x1024 as input, and outputs images of the same size.

4.2.1 Network for Coarse Confidence Maps

For producing coarse confidence maps for each body part, we use an hourglass-shaped EfficientPose B4 network, where the EfficientNet B4 backbone is implemented as described in Section 4.1.2. This network is built with five blocks, strided down to the lowest resolution in block 5 (see Figure 4.5), where it is reduced by a factor of 32 compared to the original input resolution. The feature maps are further upsampled to produce confidence maps. As seen from the figure, an input image with size 384x384 will, accordingly, produce a confidence map for each body joint of size 96 x 96. Trained on the MPII dataset, where 16 key points are labeled, the network will produce 16 different confidence maps, one for each key point. Subsequently, for HSSK, the network will produce 14 confidence map, one for each labeled key point. The network contains three bridges for forward-

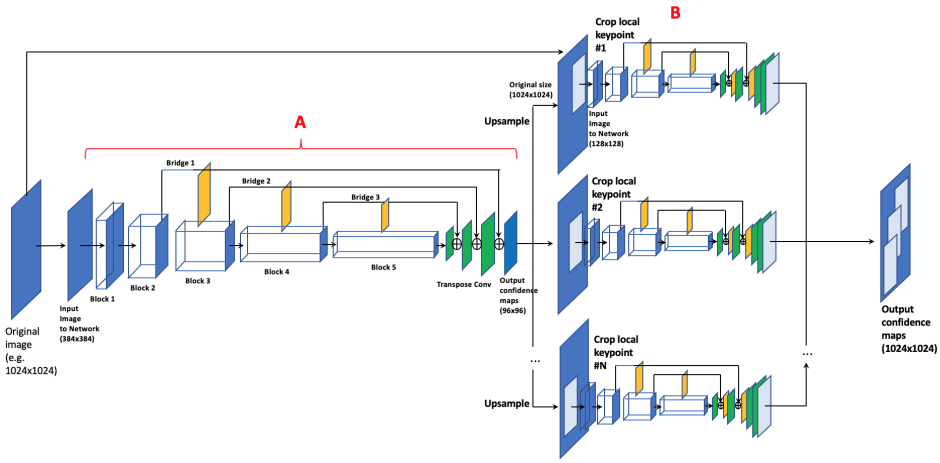

Figure 4.5: Pipeline architecture consisting of (A) EfficientPose B4 Hourglass Network and (B) EfficientPose B1 Hourglass network.

ing convolutions. As shown in Figure 4.5, one can see how these bridges are used to combine features from blocks before and after downsampling through transposed convolutions. Since the confidence maps produced by the main network acts as the input for the smaller networks, we are dependent on roughly capturing all body parts correctly. In other words, each confidence map needs to include the whole body part. If not, the network would lose crucial information, creating an impossible job for the EfficientPose B1 networks, giving the whole network a substantial drop in accuracy.

4.2.2 Network for Local Key Point Prediction

In order to produce more accurate key point predictions from the coarse confidence maps generated from the main network, we implement small hourglass-shaped EfficientPose B1 networks, where the EfficientNet B1 network is implemented as described in Section 4.1.2. The goal for implementing these networks is that by running new predictions for extremities on smaller, more efficient networks, we will be able to produce fined tuned confidence maps of higher quality, hence improving the accuracy of predicted extremities at a much lower threshold, respectively, at a PCKh measured at 10%. A single-stage hourglass network with roughly 1.4 M parameters is applied to each pair of cropped extremities produced by the main network. One single network consists of four blocks for the downsampling process. From one block to another, max-pooling reduces the feature maps with a factor of 2. Hence, with four blocks, we get the lowest resolution, reduced by a factor of 16 compared to the original input resolution. The output size is of the same magnitude as the input size, respectively, at 128x128 pixels, as shown in Figure 4.5. At each max pooling step from block number two, the network branches of, forming bridges that apply more convolutions to deeper layers. A standard hourglass-shaped network is completely symmetric, with a corresponding upsampling layer to each downsampling layer. As seen

from our model, this is not the case. We remove some of the layers for upsampling as these layers do not apply much value to the prediction.

One important aspect of this approach is that it attempts to minimize the downsampling of images. When an image is downsampled, i.e., resized to a smaller size than the original, we lose information. This information is crucial in HPE, especially when looking at lower thresholds of PCKh. By cropping out at region of interest in the original image based on the confidence maps produce by the EfficientPose B4 network and process it further in a set of EfficientPose B1 networks, we minimize downsampling while keeping a relatively high image context.

4.3 Model Exploration

In the upcoming section, we describe three different approaches used to create extremities predictions of higher quality, whereas only the approach for single body parts and segments of body parts are used in the pipeline.

4.3.1 Single Body Part

Previous network models for solving the task of Human Pose Estimation has shown a gap between accuracy for central body parts and extremities. For example, both Bulat et al. [39] and CFA [35] produces an accuracy of 98.9% and 98.7% for head predictions, but only 89.3% and 88.4% for prediction of ankles. The huge gap in accuracy between central body parts and extremities is mainly caused by the fact that extremities are more prone to occlusions, awkward poses, and crossing limbs, making the prediction of extremities a more challenging task. In other words, there is much potential for improvement in the prediction of extremities. Therefore, in this thesis, we focus specifically on extremities by only implementing smaller EfficientPose B1 networks for wrists, elbows, knees, and ankles. The single body part version with EfficientPose B1 networks focus solely on one specific body part, including both the left and right version of it.

4.3.2 Pair of Body Parts

As described in Section 2.3, humans can create a diverse set of challenging poses that might cause problems in the case of a single body part prediction. One of those is crossing limbs, creating a nearly impossible task for a model trained to not distinguish between the left and right variant of a body part (see Figure 4.6). To cope with these kinds of problems, we explore the possibility of predicting pairs of body parts. In the example of two crossing wrists, this model would be able to separate a left wrist from a right wrist, making the model robust against images containing both versions of a body part.

4.3.3 Segments

Prediction on pair of body parts works well when the limbs are close to each other, but that is not always the case. As shown in Figure 4.7, pairs of wrists, ankles, and other extremities might be positioned far from each other, resulting in unnecessary context. Hence, we

Figure 4.6: An example of a typical problem with single body part prediction.

explore models predicting segments of body parts, such as elbow and wrist, or knee and ankle. The idea is to provide more context to the network by including two connected body parts instead of one in such a way that the network can understand connections between body parts as well as the pattern of a single body part without needing too much context.

4.4 Data Processing

In order to train a neural network for the task of Human Pose Estimation, there is often required some data processing steps before the model can be presented with the data. In this section, we describe how we manipulated our initial data to a uniform structure, and most importantly, how we modified the data in order to optimize the training process.

4.4.1 Main Network

Before the CNN can use a set of images for training purposes, the images have to go through a pre-processing step. This step is necessary to ensure that all images are in the same format. We will, in this section, describe how we pre-processed the HSSK dataset.

xx First, we iterate through the annotation file to extract the annotation coordinates for the limbs of each human. At the same time, we create new objects for every individual in each of the images so that one image only represents or focus on one human at a time. In other words, if an image includes two individuals, we create two copies of the image. We include all images in the pre-processing regardless of the number of annotated limbs to get as much data as possible for the model training. This results in a total of 470,000 training images. Second, we take the original input image (Figure 4.8a) and bring the human into the center of the image using the bounding box found in the annotation file (Figure 4.8b).

a) b)

Figure 4.7: An example of a typical problem with a pair of body parts prediction. a) shows a typical human pose where the wrists are separated by a significant distance, resulting in much context when generating training data for a model based on a pair of body parts in b).

The human is centered by adding black padding to the sides of the original image, and the result can be viewed in Figure 4.8c). Third, we need all the images to have precisely the same size to ease the pre-processing part done by the model itself before training. We start by adding more padding to make the images quadratic and then change the input size of the images according to what the model expects as input. Fourth, all annotated key points need to be normalized so we can easily vary the input size of the images without having to change all annotations every time. The normalized annotations are saved in text files, one for each individual, in the same order as the given annotations in the HSSK dataset. Body parts missing annotations are given the coordinates $(0, 0)$ to prevent punishing a model who predicts body parts that are missing. The final result (see Figure 4.8d) is a dataset consisting of quadratic images in four different sizes, containing humans with at least one visible annotated key point. We divided this dataset into three duplicates whereas one was unchanged, one removed all images with one or more missing body part annotations, and the last one removed all images with missing body part annotations or annotations who were not visible.

4.4.2 Sub Network

In this sub-section, we explain how the HSSK dataset, as described in Section 2.3.1, are generated for each model approach and highlight issues that occurred during the data generation. A comparison of the three datasets can be viewed in the Figure 4.10.

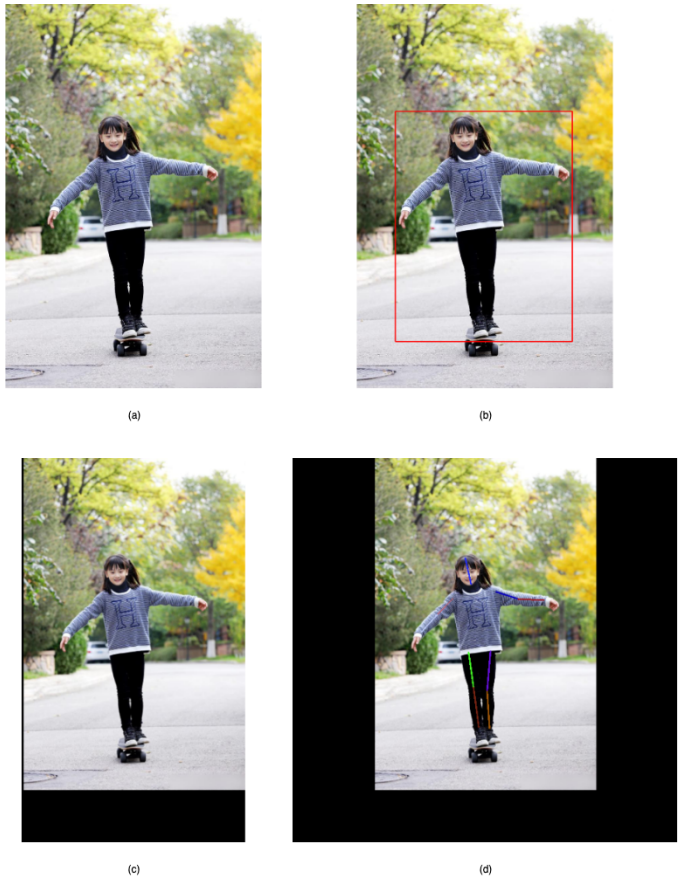

Figure 4.8: Illustration of the steps done during pre-processing of the HSSK dataset for the EfficientPose B4 network, where (a) is the original input image, (b) illustrates the bounding box of the human in the image, (c) shows how the human is centered in the image, and (d) shows the final result with a quadratic image and the corresponding human skeleton.

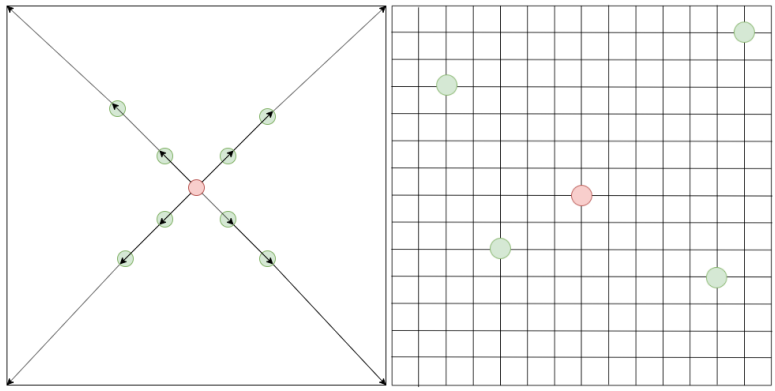

Figure 4.9: Illustration of two ways to re-position a body part within an image. a) shows our first proposed method resulting in an even distribution of points across the diagonals in the images. b) shows the final and correct method resulting in a fully random distribution of points in the images.

Single Body Part

A dataset for single body part prediction should contain images with either the left or the right version of a given body part, minimizing the number of other body parts present without reducing the context too much. Some individuals are too small in terms of size in pixels, resulting in less context than needed by the network for learning purposes. Thus we use the Euclidean distance between the head point and the neck point as a threshold to decide whether to include an image or not. As a consequence we remove all images missing annotations for the head and neck. Further, we use this scale to crop out a body part in an image giving images with a size equal to 1.5 times the head size in both directions. Now the body part is located in the center of the image. Usually, image augmentation would be done by Keras [54] during training to prevent the network from learning body part location instead of patterns, but due to unsolvable problems with Keras' image augmentation, we had to do most of it ourselves during data pre-processing. We first created a random factor based on the head size to produce a random distribution of body parts in the images, realizing that this is not a fully random distribution, but rather a distribution across the diagonals of the images as seen in Figure 4.9a). Thus, the network learned that the body part was located in those positions, and not the features of the body part itself. As a consequence, we changed the augmentation code to create two random factors based on the head size, one for each direction in the coordinate system, and used this to move the body part around to produce the fully random distribution illustrated in Figure 4.9b). To ensure that none of the points are moved out of the image or that we lose lots of context by moving it too far out in the image, we added a small safety margin to the random factors.

As with the main network, all images are made quadratic during cropping, either by including more or less of the image itself or by adding black padding. Finally, all images are resized to 128x128. The final dataset includes images of the left and the right version of the body part evenly distributed across the images in both directions and text files,

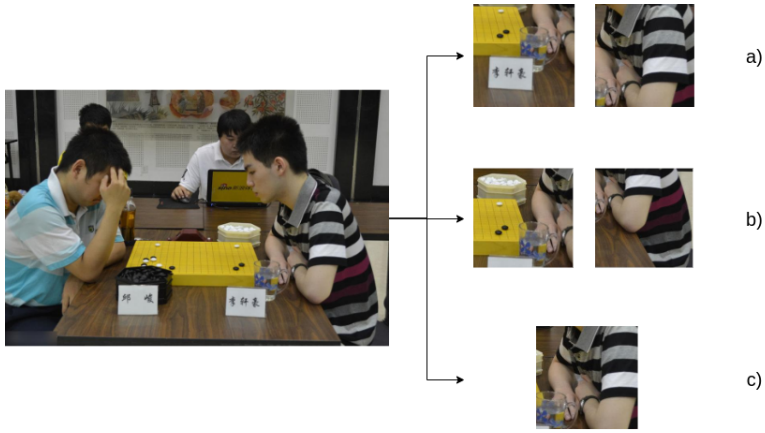

Figure 4.10: Example of an image cropped for a) single body part (wrist), b) pair of body part (wrist) and c) segment (wrist and elbow).

including the normalized annotations of the head, neck, and the body part. Since the MPII dataset is the most used evaluation dataset, we want our dataset to contain the same rate of occlusion as found in MPII. In other words, the degree of occlusion in the final dataset is set to 20%.

Pair of Body Parts and Segments

The dataset for a pair of body parts can be thought of as a segment of two body parts, the left, and the right version, respectively. Thus, the dataset for a pair of body parts and segments are generated in the same way.

Most of the process is equal to the one regarding single body parts, with a couple of exceptions. Instead of using the head size as a scaling factor in the cropping, we use the Euclidean distance between the two body parts in the segment. The cropping is done from the center point between the two body parts, resulting in a quadratic image. Then we set two random factors as in Section 4.4.2, but this time the maximum value is based on the angle between the body parts and the direction. This way, we are able to move a segment more horizontally if the segment is positioned vertically, visa versa.

The final dataset contains images of segments, including a 20% degree of occlusion based on MPII. All segments are evenly distributed around the images using the method mentioned above. Additionally, all images have one corresponding text file containing the normalized annotation for the head, neck, and the two body parts within the segment.

4.5 Training Strategy

In order to train a neural network, several aspects need to be taken into consideration, such as determination of the optimization process and augmentation strategy. In this section, we present how the optimization process and data preparation is done.

Figure 4.11: Example of the generation of confidence maps with different σ values.

4.5.1 Data Preparation

Before the images are presented to the model, we use Keras' image processing methods to apply random transformations to the images. For training of the main network, we apply random rotations of the image in a range between -45 and 45 degrees. We also add zooming, which interpolates pixel values around the image. The zoom-range spans from 0 to 25% of the image size and is uniformly randomly sampled for each dimension, that is, both width and height. In order to add even more random transformations, we apply a horizontal flip to the images, which means that some images will have their columns of pixel values reversed. Lastly, we add horizontal shifting to the image within a range of 40% of the image width. This last step is done in order to create a more robust network for scenarios where the human in focus is not completely centered in the image.

As described in Section 4.4.2, we do a lot of the data processing for the sub-network ourselves, due to unsolvable problems with Keras' image processing. However, we add some augmentation before the images are presented to the model. We use the same hyper-parameters for the training of both single body parts and segments of body parts. Since we add our own augmentation during the pre-processing of data, the images are more sensitive to scenarios where body parts of interest are moved out of the image as a result of augmentation. We therefore only add zooming that ranges from 0 to 5% of the image size, a horizontal shift in a range between 0 and 10% of the image width, and random rotations between -10 and 10 degrees when we train the sub-networks.

During the training itself, the generation of confidence maps should be implemented so that the model is rewarded for how close the prediction of a body part is to the actual ground truth value. We can modify the size of this region of the confidence maps, referred to as σ, in order to determine how strict the threshold should be for the reward. Figure 4.11 shows how the size of confidence maps for the right shoulder is changed by modifying σ, and how one can lower the criteria for reward by increasing σ. In the figure, pixel values close to 0 are expressed with purple color, while values close to 1 are expressed with yellow, indicating a strong correlation to the optimal key point location.

4.5.2 Optimization Process

The optimization process for both the main and sub-network is performed using Adam [55], which is a widely used optimization algorithm. We configure Adam with *beta 1* and *beta 2* values of 0.9 and 0.999, respectively, as suggested in Reddi et al. [55]. These values determine the exponential decay rate for the first and second-moment estimates. As addressed by Reddi et al., there exists a counter-example for non-convergence for any chosen *beta 2*. We, therefore, use the AMSGrad variation of the Adam algorithm, which allows us to use a fixed beta 2, dealing with the problem of non-convergence. Another factor to take into consideration is how much the weights should be updated after each epoch. As suggested in Reddi et al., we initiate the model using a low learning rate in order to avoid exploiting gradients. The learning rate is gradually decreased during training. As is the size of the region of the confidence map, σ. Euclidean loss is used as loss function, as defined in Equation 2.2, where *m* is the number of confidence maps, x_i is the ground truth confidence map, and t_i is the corresponding predicted confidence map. As a final trick to facilitate a stable learning process, we modify σ in different phases of the learning process, starting with a value of 10.5 for the main network and 14.0 for the sub-networks. We further decrease σ to a final value of 2.6 for the main network and 3.5 for the sub-network, which is reached in epoch 92.

4.6 Pipeline Prediction

In order to make more fine-tuned predictions of extremities, several data processing steps are required before the sub-network can produce predictions. Firstly, we need to use the predictions of the main network to find the region of interest. Secondly, we need to crop out the region of interest in such a way that the sub-network has enough context to make a prediction of high quality. Lastly, the cropped out region of interest should only contain valuable context, hence, excluding information that has no value to the prediction, or even worse, information that could make the prediction harder for the sub-network. The subsections below describe in detail how we manipulate data during the pipeline prediction for both single body parts and segments of body parts.

4.6.1 Data Flow

As mentioned in Section 4.2, the main network is presented with resized, padded images with size 1024x1024. An image processing step is required for both single body parts and segments of body parts prediction. We further feed the sub-network with the manipulated image of the region of interest. After the sub-network has made its predictions, a mapping process is required in order to convert the sub-networks predictions to its coherent predictions of the main network. The overall data flow in the pipeline can be viewed in Figure 4.12.

4.6.2 Single Body Part

Several factors are important for finding a satisfactory way to crop out the region of interest for the prediction of single body parts. The most crucial one is finding an adequate way

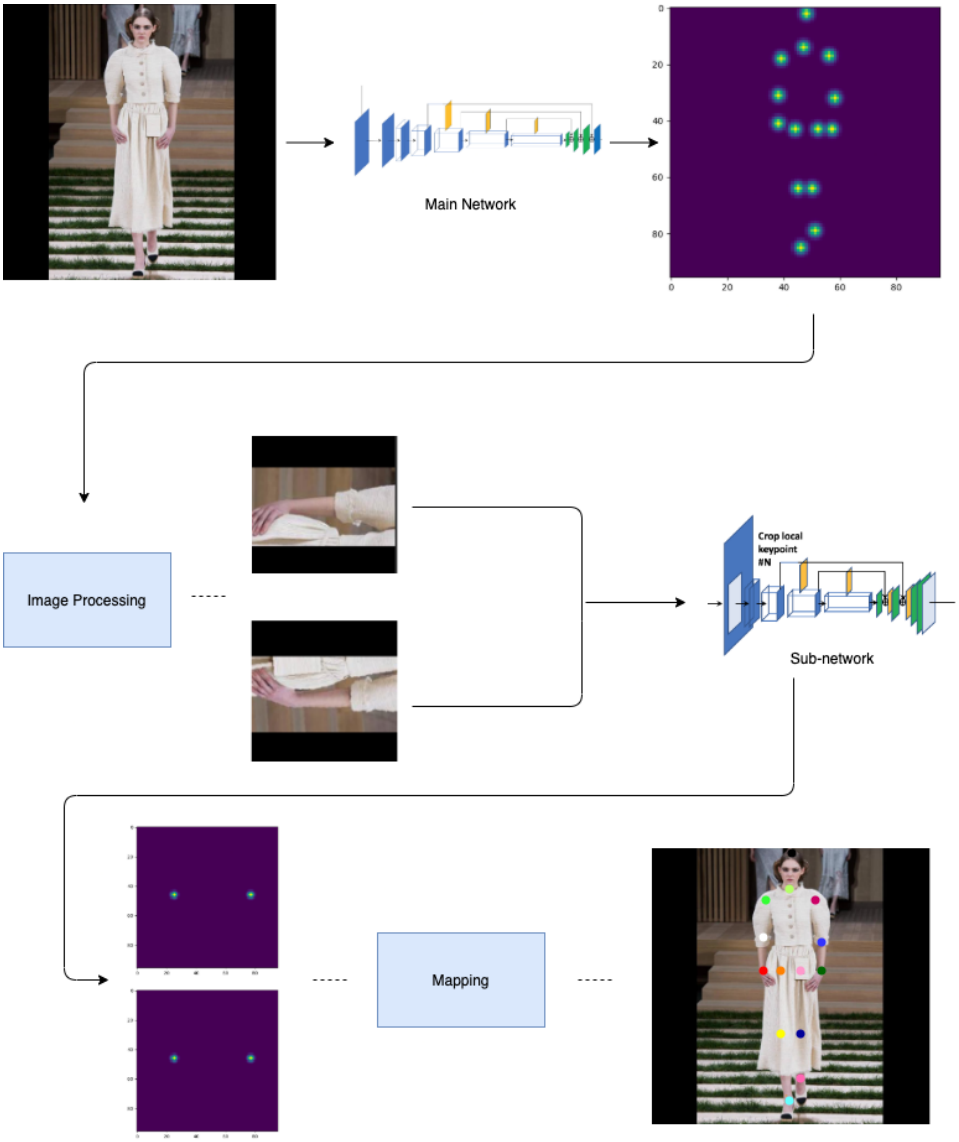

Figure 4.12: Pipeline data flow for prediction of body part segments consisting of wrists and elbows performed on the HSSK dataset.

of selecting the amount of context. We use the head bone link as a baseline threshold. By calculating the Euclidean distance between the predictions from the main network for the head top and the neck, we get the predicted size of the head bone link. We further crop out the region of interest according to this baseline threshold, such that the relevant limb is centered in the manipulated image. The cropped image is subsequently resized to 128x128 in order to make it quadratic and uniform to the expected input of the sub-network. To prevent downscaling of the input image, we set limitations to the baseline threshold. Since the sub-network uses 128x128 as input, downscaling of the input image happens when the threshold exceeds 64 pixels and is therefore limited to this value as a baseline. Another problem can occur when the human of interest is small compared to the image size. In these cases, the head bone link will be a very small value, producing a region of interest that is impossible to comprehend for the sub-network due to the lack of context. We, therefore, limit the baseline threshold value to a minimum of 25 pixels.

4.6.3 Segments

In order to solve the problem of context for segments of body parts, we first rotate the input images. The rotation makes both the process of cropping the segment and limiting the context around the segment a lot easier. For rotation, we calculate the angle in radians between the respective body parts, forming the segment according to Equation 4.1, where Δ_x represents the difference in x-direction between the first and second body part and Δ_y represents the difference in y-direction. The predicted points(x,y) produced by the main network are rotated according to Equation 4.2 and 4.3, where Δ_x represents the difference of the given x-coordinate and the x-coordinate of the center of the input image, and Δ_y represents the difference of the given y-coordinate and the y-coordinate of the center of the input image. Figure 4.13 shows the steps done during processing, where c) displays the final cropped, resized, and padded segment consisting of a wrist and an elbow. In terms of how much context the input image should contain, we first calculate the size of the head bone link by using the Euclidean distance between the top of the head and the neck. We further use 0.5 of the head bone link as a threshold value to crop out the region of interest in both x- and y-direction from the elbow and wrist. This threshold is based upon the fact that the main network should almost always make a prediction that is within 100% of the head bone link. The threshold is also limited to only 100% of the head bone link to solve the problem of crossing limbs and cases where more context can contain limbs belonging to other humans in the image. This cropping technique is further referred to as *narrow cropping*.

$$radians = \arctan(\Delta_x, \Delta_y) \tag{4.1}$$

$$rotated_x = \Delta_x \cos(r) - \Delta_y \sin(r) \tag{4.2}$$

$$rotated_y = \Delta_x \cos(r) + \Delta_y \sin(r) \tag{4.3}$$

Since the main network is considered more robust than the sub-network, we may run into cases where the main network makes the right prediction, but the sub-network makes

Figure 4.13: Illustration of the steps done during the image processing step before the prediction of segments for the sub-network. a) displays the resized, padded input image, b) is the rotated image and c) shows the cropped, resized and padded image sent to the sub-network for prediction.

a wrong prediction. As a result, the sub-network has the opposite effect to the one intended. Therefore, we use a threshold for error between the predicted limb done by the main network and the sub-network. If the Euclidean distance between the predicted limb produced by the sub-network and main network is greater than half the Euclidean distance of the head bone link, the pipeline keeps the prediction produced by the main network.

Cropping Exploration

In addition to the primary cropping technique, narrow cropping, described above, we explore two slightly different techniques to ensure maximum performance in terms of accuracy. The first one, *narrow cropping**, is based upon the same steps, but it adds a final step at the end where it rotates the cropped image back to the original position, as seen in Figure 4.14c). Since a human pose with legs horizontally is quite rare, the idea is that running prediction on a segment in the original position might be better. The second approach excludes rotation in the cropping process, which gives more context to the images in all directions (see Figure 4.14a). As a consequence, it will be more prone to images containing crossing segments or similar segments in parallel to the segment of interest. We refer to this as the original cropping method. We also explore independent cropping thresholds lengthwise and across a segment, thus the images will contain more context of the area around a segment either across the segment or lengthwise.

a) b) c)

Figure 4.14: Illustration of the three cropping techniques employed during segment prediction: a) original cropping, b) narrow cropping, and c) narrow cropping*.

Chapter 5

Results

The upcoming chapter describes the results that were gathered during the testing process based on the elaboration of the proposed method and conducted research in Chapter 4. Sections 5.1 and 5.2 focuses on isolated testing of the main network and the sub-network. Subsequently, in Section 5.3, we evaluate the pipeline and explore different modifications in the prediction process for boosting the performance.

Due to the unexpected COVID-19 situation, we did not have the opportunity to test the proposed method on infant data. The infant dataset contains sensitive data and can not leave the St. Olav's University Hospital premises. During the COVID-19 period, students were not allowed to enter St. Olav's, making the retrieval of data impossible for this project. We, therefore, throughout this chapter, only perform testing on the HSSK and the MPII dataset. More specifically, 4000 images from the HSSK test dataset and 2500 images from the MPII validation dataset. We evaluate the accuracy of the models using the standardized PCKh measurement, as described in Section 2.3.3, and the models' computational complexity measured in GFLOPS. To prevent punishing a model when body parts are missing annotations, we consider all predictions on non-annotated body parts as correct predictions.

All models used in this project are trained on the NTNU IDUN computing cluster [56] using two Nvidia Tesla V100 GPU's.

5.1 Evaluation of Main Network

The size and quality of the data are important factors in order to train a robust neural network. In addition, the size and quality are also essential in order to create tests that are both reliable and representative to the task of Human Pose Estimation. Table 5.1 shows testing results for the main network conducted on three different HSSK datasets consisting of 4000 testing images. Each test is performed in batches consisting of 1000 images per batch. (1) represents the HSSK dataset consisting of only visible, annotated body parts, (1-2) represents the HSSK dataset consisting of visible annotated body parts and non-visible body parts, and (1-2-3) is the full HSSK dataset consisting of images with visible

Model	PCKh@1	PCKh@0.5	PCKh@0.3	PCKh@0.1	PCKh@0.05
EPB4 1 on (1)	**97.7**	**91.1**	75.1	19.6	5.4
EPB4 1-2 on (1)	96.4	90.5	**76.7**	**21.2**	**5.9**
EPB4 1-2-3 on (1)	96.1	89.6	74.6	20.2	5.6
EPB4 1 on (1-2)	92.4	82.6	72.7	37.6	14.3
EPB4 1-2 on (1-2)	95.8	**88.4**	**80.7**	**47.3**	**19.0**
EPB4 1-2-3 on (1-2)	**95.9**	87.8	79.1	44.9	17.7
EPB4 1 on (1-2-3)	86.6	75.4	64.2	29.8	15.9
EPB4 1-2 on (1-2-3)	90.9	81.8	72.5	**36.3**	**18.2**
EPB4 1-2-3 on (1-2-3)	**92.8**	**83.7**	**73.9**	35.9	17.9

Table 5.1: Testing results on different HSSK dataset with 4000 test images.

body parts, non-visible annotated body parts, and missing body parts. Hence, EPB4 1-2 on (1-2-3) indicates the results for the EfficientPose B4 network trained on visible and non-visible annotated body parts tested on dataset (1-2-3). The reason why we conducted tests on several datasets was to see how big of an impact training on different data has on the test results. As we can see, EPB4 1 achieves the highest accuracy tested on (1) for both PCKh@1 and PCKh@0.5, while EPB4 1-2-3 achieves the highest accuracy for PCKh@1 tested on both (1-2) and (1-2-3). It is no surprise that EPB4 1 achieves lower accuracy when tested on (1-2) and (1-2-3) as this network has not been trained on occluded and missing body parts. What is interesting about these results is how EPB4 1-2-3 achieves the highest accuracy of the models for testing on (1-2). In this evaluation, we are mainly interested in accuracy for PCKh@1 as this measure serves as the baseline threshold for how we crop out body parts for further processing in the EfficientPose B1 models. Since EPB4 1-2-3 achieves the highest testing results in terms of accuracy on both (1-2) and (1-2-3) and based on the fact that the official MPII test set consists of both occluded and missing boy parts, EPB4 1-2-3 seems like the obvious choice for usage in the pipeline. If the model were to be trained on infant data, occluded and missing body parts would not be as crucial as for testing on the MPII dataset. This is because the videos of infants have strict rules, where videos of infants whose body parts are out of frame are considered as corrupted data.

Table 5.2 shows the testing results from EfficientPose B4 trained on (1-2-3), that is, the full HSSK dataset. As we can see, the model achieves much lower accuracy for extremities such as knees, ankles, elbows, and wrists. Furthermore, we perform tests on the same network for the MPII validation dataset in Table 5.3, both with and without re-training on the HSSK dataset.

5.2 Evaluation of Sub-Networks

In order to evaluate the sub-networks for prediction of single body parts and segments of body parts, we made a HSSK test dataset with cropped out extremities, as described in Section 4.4.2. The subsequent section presents the results of the isolated testing we conducted on the models described in Section 4.3.

	PCKh@1	PCKh@0.5	PCKh@0.3	PCKh@0.1	PCKh@0.05
Head	95.4	91.4	87.4	47.5	18.1
Neck	95.8	91.9	87.5	44.3	13.4
Left shoulder	95.6	90.3	82.0	35.6	16.6
Right shoulder	94.5	88.2	80.3	34.7	12.4
Left elbow	93.3	83.4	72.7	32.3	11.7
Right elbow	91.9	81.4	71.1	31.1	11.5
Left wrist	88.6	77.1	66.2	29.2	11.2
Right wrist	88.6	76.9	67.0	29.1	10.2
Left hip	95.3	84.9	68.3	23.7	12.5
Right hip	95.0	83.6	66.9	24.5	12.8
Left knee	92.2	81.3	70.7	38.2	25.3
Right knee	91.1	79.9	69.4	37.3	24.7
Left ankle	91.8	82.0	73.5	48.0	36.2
Right ankle	90.3	79.9	72.2	47.1	36.4

Table 5.2: EfficientPose B4 trained on full HSSK dataset (1-2-3) tested on 4000 samples from the HSSK test dataset.

	PCKh@1		PCKh@0.5		PCKh@0.1	
	Ours*	Ours	Ours*	Ours	Ours*	Ours
Head top	97.9	97.4	96.3	95.9	25.1	37.4
Shoulder	97.4	96.8	94.3	93.6	36.7	33.4
Elbows	94.8	93.5	88.7	86.5	37.5	33.5
Wrists	92.0	90.6	83.9	82.9	35.5	31.9
Hips	97.7	96.9	87.1	86.5	16.4	16.4
Knees	93.6	93.4	87.6	86.7	39.8	36.9
Ankles	92.5	91.3	86.9	85.1	50.1	48.8
Mean	95.1	94.3	90.0	89.0	34.4	34.0

Table 5.3: Test results on the MPII validation dataset. Ours* was pre-trained on the HSSK dataset.

Model	PCKh@1	PCKh@0.5	PCKh@0.3	PCKh@0.1	PCKh@0.05
EPB1 - Elbow	92.6	83.7	78.2	54.2	24.6
EPB1 - Wrist	94.2	85.7	81.1	62.0	32.2
EPB1 - Knee	88.1	71.7	64.1	36.5	14.0
EPB1 - Ankle	95.5	84.0	77.0	50.5	23.1

Table 5.4: EfficientPose B1 trained on single body parts tested on the HSSK dataset.

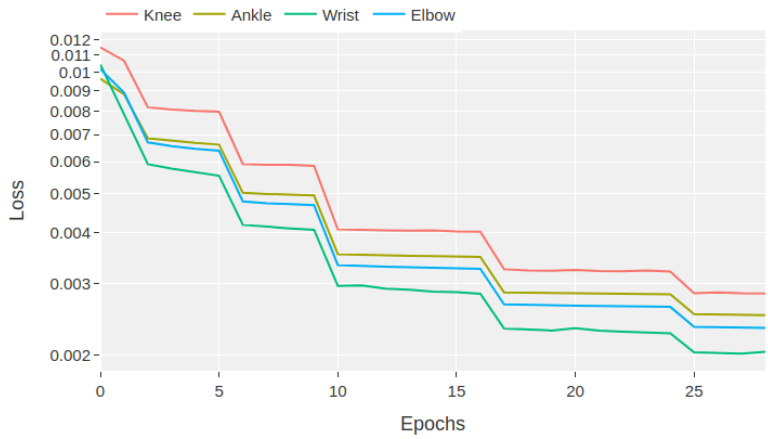

Figure 5.1: Learning curves of single body part models.

5.2.1 Single Body Part

As seen in Table 5.4, the models achieve very high accuracy for all extremities for PCKh@0.1. Compared to the main network, we see a significant increase in accuracy for elbows, ankles, and wrists, with the greatest improvement of 32.8% for predictions of wrists. We also see a substantial gap between the EfficientPose B1 model tested on knees in comparison to the EfficientPose B1 models tested on the remaining extremities.

Figure 5.1 shows the learning curves of the four single body part models for the EfficientPose B1 network. We can observe specific steps in the learning curve as a consequence of decreasing the σ according to the strategy mentioned in Section 4.5.2. The Euclidean distance between the predicted points and the ground truth of a body part during training is shown in Figure 5.2. All models perform close to equally based on the distance, except the knee model, which on average, has a 43% higher error rate than the rest.

Figure 5.3 shows predictions where our model predicts well within the thresholds of PCKh measured at 10%. We can see how the model makes predictions of high quality for both bare and clothed elbows. However, as seen in Figure 5.4, the model often struggles when there are multiple body parts in the input image. As shown in b) and c), the model chooses both the wrong elbow and knee for these images. We can clearly see a weakness of the model in these scenarios, where even though the model makes a correct prediction of an extremity, it chooses the wrong one, as the model has no prerequisite for knowing which one to choose. We can also observe how the model struggles with occluded body

Figure 5.2: Euclidean distance of single body part models during training.

Figure 5.3: Examples of successful predictions on single body parts.

parts as shown in scenario a) in Figure 5.4. As the model looses all context of the human posture when we crop out extremities, the model completely fails to estimate where the occluded knee is in the input image.

5.2.2 Segments

The overall accuracy of the two segments, elbow-wrist, and knee-ankle, can be viewed in Table 5.5. Compared to the result of the EfficientPose B4 model on the corresponding body parts, we can see a significant improvement for all thresholds. In wrist prediction alone, we can see an improvement of over 50% in terms of accuracy on the PCKh@0.1 measure.

Model	Body part	PCKh@1	PCKh@0.5	PCKh@0.3	PCKh@0.1	PCKh@0.05
EPB1 - Elbow-wrist	Elbow	98.6	96.8	94.5	74.6	40.2
	Wrist	98.8	97.3	95.1	79.4	46.0
EPB1 - Knee-ankle	Knee	94.1	86.3	81.4	45.8	51.8
	Ankle	94.2	87.2	82.3	51.8	22.2

Table 5.5: Testing results for EfficientPose B1 trained on segments and tested on the HSSK dataset.

Figure 5.4: Examples of failed predictions on single body parts.

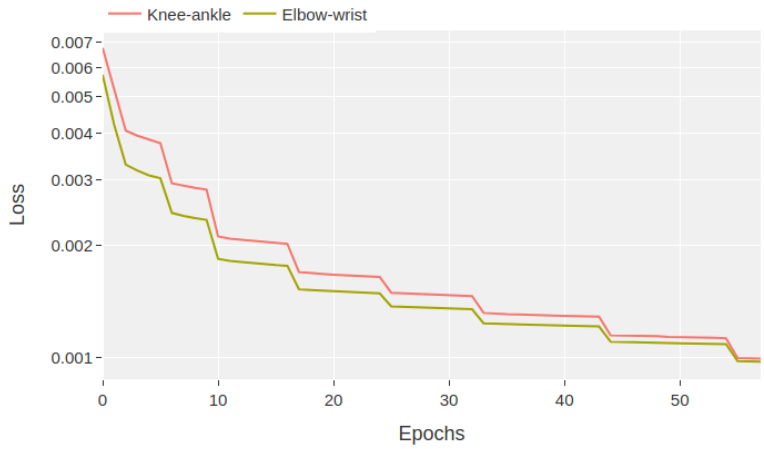

Figure 5.5: Learning curves of segment models.

Figure 5.5 shows the same tendency in learning rate as for single body part where the curves follow the decreasing σ value. Both models achieve relatively low loss values and low error rates, as shown in Figure 5.6. Again, we observe that the knee poses as a challenge in the segment prediction with a higher Euclidean distance than for the elbow-knee segment, as in the single body part prediction.

As seen in Figure 5.7 the two models for segment prediction achieve high accuracy on non-occluded body parts both on bare body parts and clothed body parts as long as the image provides enough texture in the clothing. Thus, if a person wears a piece of clothing that is straight and smooth and covers the body part, as in Figure 5.8b), the model struggles to uncover the patterns of it. The same problem occurs in Figure 5.8d) where the wrist is occluded. Some human poses, like the ones shown in Figure 5.8a) and c), turn out to be difficult to predict correctly for the segment models due to crossing or similar body parts within one image. This is a general problem for both the single body part and segment sub-networks due to the limited image context.

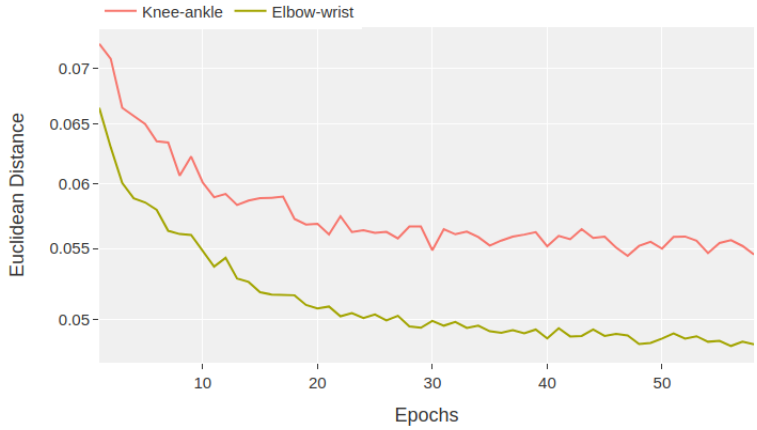

Figure 5.6: Euclidean distance of segment models during training.

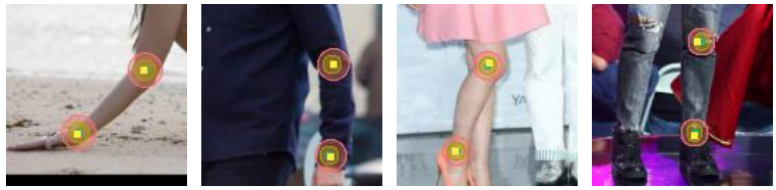

Figure 5.7: Examples of successful predictions on segments of body parts.

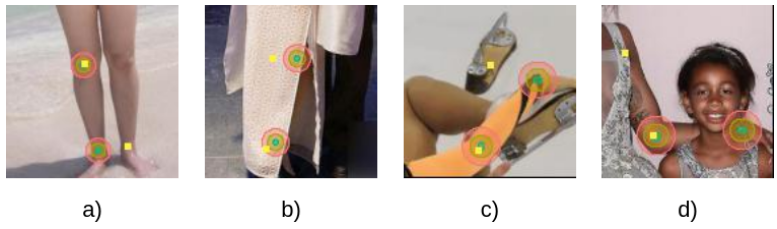

Figure 5.8: Examples of failed predictions on segments of body parts.

	54	64	80	90	100
Left elbow	27.8	31.6	**35.2**	33.4	35.0
Right elbow	27.4	30.7	**33.5**	33.4	32.5
Left wrist	27.1	30.0	**32.4**	**32.4**	31.9
Right wrist	27.3	30.0	32.9	**33.2**	32.3
Left knee	29.5	31.0	33.0	33.2	**33.4**
Right knee	29.9	32.5	**35.1**	34.9	35.0
Left ankle	45.0	47.4	**50.2**	50.0	49.7
Right ankle	45.2	48.0	**51.3**	51.2	50.6

Table 5.6: Testing of different max-scale thresholds given in pixels for prediction of single body parts. The results are given for PCKh@0.1.

5.3 Evaluation of Pipeline

Throughout this section we show tests carried out on the implemented pipeline. These tests were conducted in order to optimize the performance of the pipeline. All tests are performed on 2500 images from MPII's validation dataset. The final testing is conducted on MPII's official test dataset, which contains 7400 images.

5.3.1 Single Body Parts

As mentioned in Section 4.6.2, we set a baseline threshold for cropping out single body parts based on the predictions of the main network. Even though the maximum baseline value for preventing downscaling is 64 pixels, we conducted tests with different thresholds to see the difference in the performance of the model. Table 5.6 shows performance of EfficientPose B1 trained on the HSSK dataset for different max-scales measured at PCKh@0.1. As seen, surprisingly, the model performs best when we limit the cropping threshold to 80 pixels. This shows that even though some pictures are downscaled, the trade-off from providing the model with more context pays off.

5.3.2 Segments

We have tested several methods for prediction of segments of body parts in the pipeline. As described in Section 4.6.3, the most important factor is finding a satisfactory way of cropping out the segments of interest based on the main network's predictions. This subsection shows the conducted testing for different ways of cropping out segments of body parts, and how usage of multiple datasets affects the prediction accuracies.

Exploration of Cropping Context and Cropping Method

We explore different thresholds for cropping out the segment of interest. As described in Section 4.6.3, we use half the size of the head bone link as a baseline threshold for cropping. Table 5.7 shows the conducted experiments for different thresholds. We also explore the usage of different cropping techniques on the images fed to the sub-networks, all described in Section 4.6.3. We see that the original cropping method outperforms both

Cropping method	Body part	0.15	0.25	0.3	0.33	0.35	0.4	0.5	0.55	0.6
Narrow cropping	Elbow	13.0	19.6	**21.0**	20.5	20.2	18.0	12.0	9.1	6.9
	Wrist	22.7	27.9	**28.3**	28.2	28.1	26.7	20.3	17.1	14.6
	Knee	19.9	20.9	21.2	21.3	21.5	21.9	22.2	**22.3**	21.7
	Ankle	36.6	38.4	39.1	39.6	**39.8**	39.6	39.0	38.3	37.3
Narrow cropping*	Elbow	-	-	25.2	**25.9**	25.1	23.6	17.4	15.0	12.5
	Wrist	-	-	**31.6**	**31.6**	30.9	29.1	23.1	19.4	17.3
	Knee	-	-	23.7	24.2	24.5	25.6	28.3	29.0	**29.7**
	Ankle	-	-	43.8	43.5	44.0	44.9	**45.2**	44.6	43.5
Original	Elbow	-	-	27.6	28.6	**29.1**	27.5	21.6	18.4	14.8
	Wrist	-	-	32.1	**32.8**	32.5	31.8	26.2	23.1	20.3
	Knee	-	-	24.5	24.9	25.1	26.3	28.8	30.0	**30.6**
	Ankle	-	-	43.9	44.8	45.1	46.0	**46.9**	46.8	46.1

Table 5.7: PCKh@0.1 across segment body parts for different cropping thresholds and cropping techniques.

	(0.365, 0.5)	(0.5, 0.5)	(0.625, 0.5)	(0.75, 0,5)
Left elbow	21.64	**22.36**	18.96	12.64
Right elbow	16.76	**18.64**	16.36	10.32
Left wrist	27.40	**28.00**	25.52	20.32
Right wrist	27.40	**28.40**	27.56	21.76
Left knee	20.40	21.40	22.12	**22.48**
Right knee	20.24	21.20	22.60	**23.44**
Left ankle	38.08	**39.60**	38.84	38.27
Right ankle	38.84	39.60	**39.92**	36.28

Table 5.8: PCKh@0.1 across segment body parts for various cropping thresholds in two directions. The thresholds are defined as (x, y) where x is the threshold lengthwise with the segment and y is across.

methods with narrow cropping on all body parts. It can also be observed that the knee-ankle segments need more image context than the elbow-wrist segment to maximize its performance, regardless of the employed cropping method. Since cropping thresholds below 0.3 resulted in poor performance, we chose not to conduct similar experiments on the remaining methods. Further on, we explore how the accuracy responds to varying cropping thresholds in the horizontal and vertical direction in Table 5.8, a test that only applies to the narrow cropping method. Results show that we achieve the highest overall accuracy using the same threshold in both directions.

Exploration of Multiple Datasets

Based on the performance boost in accuracy of exploiting both the HSSK and the MPII dataset in training, as shown by Bulat et al. [39] and Su et al. [35], we explore this very concept for prediction of segments in the pipeline. We, therefore, conduct tests for EfficientPose B1 models trained on HSSK and MPII separately and compare these results to an EfficientPose B1 model trained on both datasets. Table 5.9 shows the difference in

Model	Body part	PCKh@1	PCKh@0.5	PCKh@0.3	PCKh@0.1	PCKh@0.05
EPB1 - Elbow-wrist	Elbow	90.8	77.5	66.0	29.0	10.3
	Wrist	90.9	77.4	67.3	32.4	13.9
*EPB1 - Elbow-wrist	Elbow	92.12	82.0	71.8	36.0	14.3
	Wrist	90.2	79.2	70.0	37.3	16.2
**EPB1 - Elbow-wrist	Elbow	90.5	76.1	65.0	28.9	10.6
	Wrist	88.7	75.8	65.1	32.0	13.0
EPB1 - Ankle-knee	Ankle	89.1	78.6	70.0	46.1	36.4
	Knee	83.2	68.8	56.1	30.6	21.8
*EPB1 - Ankle-knee	Ankle	89.0	80.9	73.9	48.3	37.4
	Knee	82.5	70.9	60.7	32.4	22.3
**EPB1 - Ankle-knee	Ankle	90.5	81.8	73.7	48.4	37.8
	Knee	85.2	72.4	62.0	33.6	23.1

Table 5.9: EfficientPose B1 model trained on segments with different datasets and tested on the MPII dataset. EPB1 is only trained on the HSSK dataset, *EPB1 is only trained on the MPII dataset, and **EPB1 is trained on both HSSK and MPII.

	0.1	0.15	0.2	0.3	0.5	0.55	0.6
Elbow	38.2	38.0	37.5	35.9	36.0	35.9	35.8
Wrist	37.1	37.4	37.4	36.6	36.2	36.1	36.1
Knee	39.6	39.1	38.0	37.2	35.7	35.6	48.8
Ankle	49.9	49.8	49.7	49.7	49.1	49.0	35.3

Table 5.10: Comparison of different thresholds used for deciding when to keep predictions made by the sub-network or not. The results are given for PCKh@0.1.

performance for each model.

Exploration of Merged Predictions

As described in Section 4.6.3, we may run into cases where the main network makes the right prediction, but the sub-network makes a wrong prediction. We, therefore, explore different thresholds for when to keep predictions made by the sub-network or not. As seen from Table 5.10, the pipeline perform best if we only keep predictions made by the sub-network when the Euclidean distance between the main and sub-network prediction is less than 10% of the size of the head bone link.

Final MPII Results

To fully optimize our method, we propose a final hybrid solution consisting of segment prediction for elbows and wrists combined with a single body part prediction for knees and ankles, as this combination shows the most promising results. Table 5.11 shows a comparison of the main network and the final hybrid solution for our pipeline tested on the MPII validation dataset. We can see an overall increase in precision of 0.9% compared to the main network, where the final pipeline scores best for three out of four body parts.

To obtain an official evaluation for the MPII dataset, we submitted our predictions on the MPII test dataset to Max Planck Institute for Informatics, stationed in Germany.

	Main Network	Pipeline
Elbows	37.5	**38.2**
Wrists	35.5	**37.1**
Knees	**39.8**	39.6
Ankles	50.1	**51.6**
Mean	40.7	41.6

Table 5.11: Comparison between the main network and the pipeline predictions for extremities measured at PCKh@0.1 on the MPII validation dataset.

	Ours	Bulat et al.	Newell et al.	Wei et al.
Elbows	35.8	**44.2**	39.3	39.0
Wrists	36.9	**43.6**	37.2	36.8
Knees	25.6	**33.6**	28.7	27.1
Ankles	29.0	**34.1**	29.4	29.0

Table 5.12: Comparison between our method and current state-of-the-art methods for extremities measured at PCKh@0.1 on the MPII test dataset.

Table 5.12 shows the testing results for our method in comparison to other state-of-the-art methods.

5.4 Runtime Performance

When it comes to measuring the efficiency of the EfficientPose models, we evaluate our models against models achieving state-of-the-art results, both in terms of the size of the network and the computational requirements for making predictions of high quality. Figure 5.9 shows a comparison between our models and current state-of-the-art models in terms of GFLOPS, while Figure 5.10 compares the size of the networks. We see that the architecture of EfficientPose B4 is both significantly lighter and quicker than the models it has been evaluated against. In fact, compared to CFA [35] we see a computational reduction of 85.5% and a 69.5% reduction compared to HPE [39]. We also observe that EfficientPose B4 is 38% lighter than CFA and 31% lighter than HPE in terms of the number of parameters used in the networks.

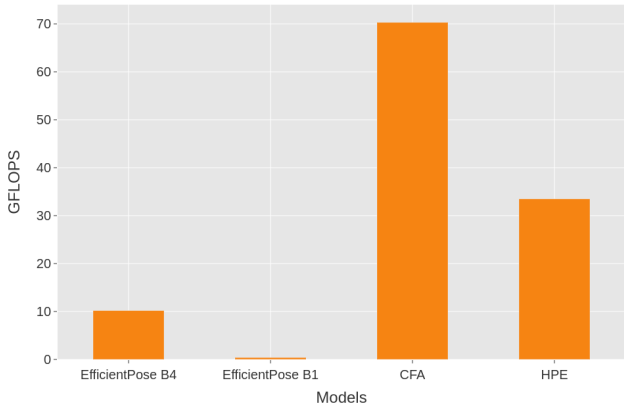

Figure 5.9: Comparison of GFLOPS between our models and CFA [35] and HPE [39].

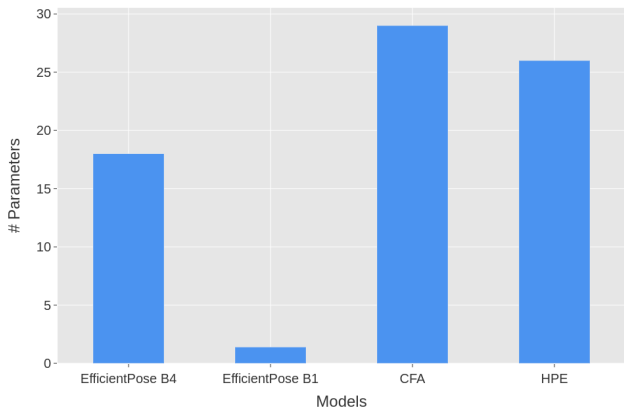

Figure 5.10: Comparison of number of parameters used between our models and CFA [35] and HPE [39].

Chapter 6

Discussion

In the upcoming chapter, we interpret and reflect upon the results presented in Chapter 5. We highlight challenges that occurred during this project, as well as the advantages and disadvantages of the proposed method. We begin with a discussion of the annotation quality of the exploited data in Section 6.1. Then Section 6.2 compares the implemented methods used for prediction of extremities, while Section 6.3 addresses the exploration of image context for these methods. Subsequently, Section 6.4 discuss the limitations and weaknesses of our method, and lastly, Section 6.5 answers the research questions in light of the work conducted in this project.

6.1 Annotation Quality of HSSK and MPII

In this project, we have exploited two large datasets for the task of HPE, namely MPII and HSSK. The amount of training data has been crucial to achieve satisfactory results, and we have observed that most models benefit from the increased amount of training data. However, we have seen that the mix of two datasets has given a precision downgrade in some cases. The model for the elbow-wrist segment gave significantly better results when trained on the MPII alone compared to being trained on HSSK and re-trained on MPII (Table 5.9), especially in terms of lower PCKh thresholds. We have also experienced visible differences in the annotation strategies of several body parts in the HSSK and MPII datasets, which strongly suggests that this has been the reason for the precision drop.

Since most state-of-the-art methods on the MPII dataset have had a goal to maximize accuracy on the PCKh@0.5, the small differences in annotations between HSSK and MPII have been well within the threshold, thus combining the two have given the methods a final boost in terms of accuracy. A small error in an annotation of a body part in medical purposes like CP prediction could be critical to the final outcome. In other words, high precision in terms of PCKh@0.1 or lower would be of great importance, and annotation consistency in the dataset annotations is crucial to satisfy this requirement, a necessity neither HSSK nor MPII provide.

6.2 Single Body Part vs Segment Prediction

In Chapter 5, we explored several solutions for making predictions of higher quality for extremities. Section 5.2 showed isolated testing of the EfficientPose B1 models for single body parts and segments of body parts. We found that the approach for segments of body parts outperforms the approach for single body parts. The most significant improvement was seen for the prediction of wrists, where segments of body parts achieve 79.4% for PCKh@0.1 compared to 62% for prediction of single body parts. The approach for segments of body parts achieved higher accuracy for all extremities, that is, ankles, wrists, knees, and elbows. These results confirmed our theory as stated in Section 4.3.3, that by providing more context to the network by including two connected body parts instead of one in such a way that the network could understand connections between body parts, the network would perform better. However, integrated testing for both approaches in the pipeline showed less clear results. Surprisingly, the approach for single body parts performed best for ankle and knee prediction, compared to the approach for segments. Therefore, a hybrid solution between the single body part approach and the segment approach seemed most beneficial when integrated into the pipeline.

6.3 Exploration of Image Context

6.3.1 Single Body Part

Aiming to optimize the performance of the pipeline, Section 5.3.1 compared different thresholds for cropping out the body part of interest. As mentioned in Section 4.6.2, our initial thought was to limit the minimum cropping value to 25 pixels if the size of the head bone link was smaller than this value. The rationale for this boundary was to comprehend problems that could occur when the human of interest was small compared to the image size. In these cases, the head bone link would be a very small value, producing a region of interest that would be impossible to comprehend for the sub-network due to the lack of context. Despite this, testing carried out on different minimum thresholds for cropping out single body parts showed no difference in prediction accuracy. We, therefore, chose to exclude further testing for minimum thresholds, as we saw that only the maximum value for the cropping threshold made any difference to the prediction accuracy. Testing results for maximum cropping thresholds (Table 5.6) showed that the EfficientPose B1 models achieved the best accuracy for almost every measure when the maximum value was set to 80 pixels. This is an interesting result, as we would expect that the highest accuracy would be achieved when we limit the maximum value to 64 pixels, as this prevents downscaling of the image. Hence, even though some images are downscaled, providing the network with information of lower quality, the trade-off by providing the network with more context is more beneficial.

6.3.2 Segments

As described in Section 5.3.2, we explored three different cropping strategies as an attempt to optimize the performance of the model. Our main theory for the primary cropping

technique, *narrow cropping*, as described in Section 4.6.3, performed surprisingly poor compared to our two exploratory cropping techniques (Table 5.7). In fact, the cropping technique that enhanced the sub-networks performance the most was the *original* cropping technique, where we completely discarded the concept of rotation. We believe that this comes as a direct result of the way we pre-processed the data used during the training process. Since the *original* cropping technique feeds the sub-network with data of the exact format as the training data, it makes sense that this technique enhances the performance of the sub-network the most. However, we could also have processed the training data in a similar fashion to the *narrow cropping* and *narrow cropping** techniques in order to increase the results of these approaches. As training is a time-consuming task, we did not get the chance to train the networks with that kind of data. As an attempt to further utilize the cropping techniques, we conducted testing for different cropping thresholds (Table 5.7), where the results were less clear. For segments consisting of ankles and knees, it appeared that a cropping threshold between 50-60% of the size of the head bone link gave the best results in terms of accuracy, but for segments consisting of wrists and elbows, a cropping threshold between 30-35% of the size of the head bone link seemed like the optimal value.

As a final attempt to optimize the quality of predictions for segments of body parts using the *narrow cropping* technique, we explored how the accuracy responded to varying cropping thresholds in the horizontal and vertical direction (Table 5.8). The results showed that there was no performance boost in terms of accuracy for varying cropping thresholds in the horizontal and vertical direction; on the contrary, the network performed worse. In other words, the network performed best when the segment was cropped out with equal values in both directions.

6.4 Limitations and Weaknesses

Throughout the testing conducted in Section 5.3, we revealed both limitations and weaknesses with our method. First of all, it was clear from the very beginning that the quality of the sub-networks' predictions was highly dependent on the performance of the main network. As shown in Table 5.3, the overall performance of the EfficientPose B4 model on extremities tested on MPII validation dataset was 92.9% for PCKh@1. As a consequence, the sub-network will be fed with many cropped out images that do not contain the body part of interest. As we have no ground truth values for predictions made by the main network on real-world data, we have no basis for implementing a smarter solution to comprehend this problem. The only way of preventing the sub-network from receiving images without the body part of interest would be to use a network that makes no error on real-world data. Unfortunately, this is not possible as there exists no such network.

During the testing process for the implemented pipeline, we made visual inspections of the sub-network predictions and compared them to the coarse predictions made by the main network. The visual inspections showed a drop in prediction performance in the sub-network for occluded body parts. It became clear that one of the most significant drawbacks of our method was predictions on occluded body parts, as the model loses all context of the human posture when we crop out extremities (Figure 5.8b)). An intelligent implementation, where the network can detect occlusions and use the predictions from the

main network instead of the predictions made by the sub-network, is therefore needed to solve the issue of occluded body parts.

Another limitation of the method that comes as a direct consequence of the loss of context is the separation between crossed limbs and limbs that appears in a parallel position, shown in Figure 5.8 and 5.4. As mentioned in Section 4.3.3, we implemented a new way of cropping out segments to overcome this challenge. However, this narrow cropping method achieved lower accuracy in general (Table 5.8). We, therefore, concluded that the best solution would be to prune out wrong predictions from the sub-network by only keeping predictions when the Euclidean distance between the main and sub-network prediction was less than 10% of the size of the head bone link.

As MPII has become the standard dataset for measuring the quality of a method for Human Pose Estimation, it was evident that we needed to use this dataset as a test set. However, the MPII dataset consists of many low-quality images, as it is made from scraped YouTube videos from 2014. Since our pipeline is based on images of 1024x1024 as input, we need to upscale the images from the MPII dataset that are of a lower resolution, resulting in low-quality input for the cropped out images in the sub-network. We, therefore, argue that our model will achieve better results on infant data, as it contains images of higher resolution and data of a more standardized format.

6.5 Answering Research Questions

As stated by the research questions, this project was carried out to evaluate whether it was possible to produce predictions of higher quality merely by modifying the network architecture and exploiting more data. With extremities as main focus, we discuss the research questions based on the key findings in Chapter 5.

1. How can the task of Human Pose Estimation be optimized to produce predictions of higher quality for cerebral palsy?

This project was based upon the fact that the prediction quality of a higher precision was hard to accomplish with a network trained to perform full-body prediction. Current state-of-the-art methods for HPE have focused solely on PCKh@0.5, whereas within the field of medicine, a metric of higher precision is required. Our method achieves an overall increase of precision for PCKh@0.1 of 0.9% for extremities on the MPII validation set. Due to the unexpected COVID-19 situation, we did not get the chance to verify whether the increase in accuracy also applies to Infant Pose Estimation.

1.1. How can we modify the network architecture to produce higher overall accuracy for predicted body parts measured at lower thresholds?

To achieve the goal of an increase in the overall accuracy measured at lower thresholds, we proposed a two-staged network architecture combining a network performing full-body prediction and several light-weight sub-networks specialized for increasing the prediction

quality of one specific body part. This single body part approach gave a significant increase in precision in isolated testing on the HSSK dataset, which strengthened the hypothesis. However, further testing combined with the main network showed that the approach performed worse than the main network alone.

To enhance the proposed method, we refined the sub-networks by introducing segment prediction, providing more context to the sub-networks without increasing the number of parameters. As a result, we were able to slightly increase the performance of the combined architecture for both the elbows and the wrist, but the knees and ankles still posed as a problem. Finally, a hybrid solution consisting of segment prediction for elbows and wrists combined with a single body part prediction for knees and ankles gave the best results with an overall increase in precision for PCKh@0.1 of 0.9% for extremities on the MPII validation dataset. Based on these results, we have confirmed that a two-staged network architecture can increase the accuracy of a Human Pose Estimation model, when focusing on thresholds of PCKh@0.1 or lower.

The final test completed on the MPII test dataset showed that our proposed method produced lower results than current state-of-the-art on the PCKh@0.1 (Table 5.12). Due to a low number of parameters on our EfficientPose B4 model compared to those methods, we argue that a larger version in terms of parameters would be able to compete with the current state-of-the-art. Nevertheless, the performance boost of adding several smaller EfficientPose B1 models for extremity prediction has shown to be relatively low, thus adding them to this larger model could require a high effort compared to the reward.

1.2. How can we increase key point accuracy of the model merely based on exploiting available data?

The MPII dataset has been used as the primary source for training and benchmarking of neural networks for Human Pose Estimation due to its large amount of annotated images for both single- and multi-person pose estimation. Lately, methods have made use of multiple datasets to increase its performance, resulting in state-of-the-art results on the MPII test set for the PCKh@0.5 metric. In combination with the proposed two-staged network architecture, we have explored how and whether a second dataset, namely the HSSK dataset, could be used to increase precision for more accurate metrics like the PCKh@0.1. Starting with the EfficientPose B4 network, we observed a small overall increase in PCKh@0.1 when making use of both datasets during training. However, some specific body parts got a significant drop in accuracy, like the head top, which dropped from 37.4% to 25.1%. We saw the same tendency when training the EfficientPose B1 models based on segment prediction. The elbow-wrist segment dropped from 36.7% to 30.5% on PCKh@0.1 when pre-trained on HSSK, while the knee-ankle segment improved. We argue that small differences in body part annotations between the two datasets result in a performance drop when combined, primarily for more accurate metrics.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

Body tracking is one of many tasks necessary for a computer-based system for CP prediction. However, accurate predictions of body parts are essential for producing CP predictions of high quality. Current state-of-the-art methods for Human Pose Estimation have focused on a precision threshold that does not meet the standards required for high-quality prediction of CP. We have presented how medical experts diagnose infants with CP, and how we can apply Computer Vision and CNN's to support this important job. Employing the medical and technical theory presented in this thesis, we have proposed a two-staged modification of a convolutional neural network architecture that can be used to increase the prediction quality of extremities, measured at lower thresholds.

As recent state-of-the-art methods within the field of Human Pose Estimation have shown an increase in key point accuracy by exploiting more available data, we have explored the usage of multiple datasets. We have observed that exploiting more data during training may increase the accuracy of the model for higher thresholds. However, for lower thresholds, we have in some cases observed a drop in accuracy. We, therefore, conclude that the performance of a model exploiting more data depends on the annotation consistency, as an inconsistency in the annotations will weaken the benefits of an increased amount of data.

We have explored how the usage of multiple light-weight sub-networks in combination with a more extensive network for coarse body part predictions can increase the overall accuracy for predictions of extremities. Furthermore, we have proposed several ways to implement these light-weight sub-networks, using the sub-networks for predictions of both single body parts and segments of body parts. The method suffers from some limitations, in particular for the prediction of occluded body parts, as we lose context of the human posture when performing isolated predictions for extremities. With an overall increase of 0.9% compared to the main network, we can conclude that our modification of the network architecture can be used to increase the overall precision for predicted key points measured at lower thresholds. However, one can argue that the high effort of adding several light-

weight networks has shown little reward in performance.

7.2 Future Work

With the work conducted in this project and the aspects discussed in earlier chapters, we propose four natural steps that could be further explored to optimize the method.

To fully utilize the narrow cropping techniques, it seems most beneficial that the models are trained on data that are cropped out using the same technique. Subsequently, this means generating new datasets for training purposes where the segments of interest are cropped out using the narrow cropping technique, and lastly, conducting training on the sub-networks from scratch on the newly generated datasets.

As stated earlier, one of the most significant drawbacks of our method is the prediction of occluded body parts. Based on this limitation, we suggest that one should explore a more sophisticated solution for detecting occluded body parts. If the network could detect occluded body parts, we could choose to keep the main network's predictions based on whether the body part was occluded or not. Such a solution would most likely increase the overall quality of the predictions as it would utilize the strengths of both networks.

Due to the unexpected COVID-19 situation, we did not get the opportunity to retrieve the infant children data stored at St. Olav's. Since our goal was to improve the quality of CP-predictions by providing extremity predictions of higher quality, conducting testing for infant children's data would be crucial for further development of our method. Hence, both training and testing on infant children's data are required steps for this project.

As one last exploratory step, we suggest the implementation of an object-detector for localizing the body parts of interest in order to improve the detection rate. In that way, the input to the specialized sub-networks would be determined either by the output of the object-detector alone or in combination with the output of the main network. The object-detector would then further pass on the detected segment to the sub-network for a prediction of each body part in the segment.

Bibliography

[1] B. Larroque, P.-Y. Ancel, S. Marret, L. Marchand, M. André, C. Arnaud, V. Pierrat, J.-C. Rozé, J. Messer, G. Thiriez, A. Burguet, J.-C. Picaud, G. Bréart, and M. Kaminski, "Neurodevelopmental disabilities and special care of 5-year-old children born before 33 weeks of gestation (the epipage study): a longitudinal cohort study," *EPIPAGE Study group*, 2008.

[2] P. Rosenbaum, N. Paneth, A. Leviton, M. Goldstein, and M. Bax, *The definition and classification of cerebral palsy*. Dev Med Child Neurol, 2007.

[3] L. Adde, J. Helbostad, A. Jensenius, G. Taraldsen, K. Grunewaldt, and R. Støen, "Early prediction of cerebral palsyby computer-based video analysis of general movements: a feasibility stud," 02 2010.

[4] M. Bosanquet, L. Copeland, R. Ware, and R. Boyd, "A systematic review of tests to predict cerebral palsy in young children," *Developmental medicine and child neurology*, vol. 55, pp. 418–26, 05 2013.

[5] L. Adde, R. Støen, M. Rygg, K. Lossius, and G. Øberg, "General movement assessment: Predicting cerebral palsy in clinical practise," vol. 83, 01 2007.

[6] N. Hesse, C. Bodensteiner, M. Arens, U. G. Hofmann, R. Weinberger, and A. S. Schroeder, "Computer vision for medical infant motion analysis: State of the art and rgb-d data set," 2018.

[7] P. Rosenbaum, R. Palisano, S. Walter, E. Wood, and B. Galuppi, *Development and reliability of a system to classify gross motor function in children with cerebral palsy*. Dev Med Child Neurol, 1997.

[8] P. Rosenbaum, R. Palisano, D. Barlett, and M. Livingston, "Gross motor function classification system expanded and revised," 2007.

[9] C. P. alliance, "Gross motor function classification system (gmfcs)." `https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/severity-of-cerebral-palsy/gross-motor-function-classification-system/`, 2018.

[10] M. Hadders-Algra, "General movements: a window for early identification of children at high risk for developmental disorders," 2004.

[11] C. Einspieler, R. Peharz, and P. B. Marschik, "Fidgety movements. tiny in appearance, but huge in impact," *Jornal de Pediatria*, vol. 92, pp. 64 – 70, 06 2016.

[12] H. Prechtl, C. Einspieler, G. Cioni, A. Bos, F. Ferrari, and D. Sontheimer, "An early marker for neurological deficits after perinatal brain lesions," 1997.

[13] C. Einspieler and H. Prechtl, "Prechtl's assessment of general movements: A diagnostic tool for the functional assessment of the young nervous system," 2005.

[14] C. Einspieler and H. Prechtl, "Prechtls assessment of general movements: A diagnostic tool for the functional assessment of the young nervous system," 2005.

[15] M. Hadders-Algra, "The assessment of general movements is a valuable technique for the detection of brain dysfunction in young infants. a review," acta paediatrica (oslo, norway : 1992)," 1996.

[16] K. Lorenz, "Gestalt perception as a source of scientific knowledge," 1971.

[17] C. Einspieler, H. Prechtl, and F. Ferrari, "The qualitative assessment of general movements in preterm, term and young infants - review of the methodology," 1997.

[18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall Press, 3rd ed., 2009.

[19] F. Chollet, *Deep Learning with Python*. Greenwich, CT, USA: Manning Publications Co., 1st ed., 2017.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[21] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017.

[22] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," *CoRR*, vol. abs/1906.11172, 2019.

[23] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *CoRR*, vol. abs/1812.08008, 2018.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[25] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[27] L. Sigal, *Human pose estimation. In Computer Vision*. Springer, 2014.

[28] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," June 2014.

[29] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[30] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, Y. Wang, and Y. Wang, "AI challenger : A large-scale dataset for going deeper in image understanding," *CoRR*, vol. abs/1711.06475, 2017.

[31] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," 2019.

[32] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," 2019.

[33] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, *Human pose estimation using body parts dependent joint regressors*. 2014.

[34] A. Bearman, Stanford, and C. Dong, "Human pose estimation and activity classification using convolutional neural networks," 2015.

[35] Z. Su, M. Ye, G. Zhang, L. Dai, and J. Sheng, "Improvement multi-stage model for human pose estimation," *CoRR*, vol. abs/1902.07837, 2019.

[36] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *CoRR*, vol. abs/1603.06937, 2016.

[37] L. Ke, M. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," *CoRR*, vol. abs/1803.09894, 2018.

[38] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *CoRR*, vol. abs/1901.00148, 2019.

[39] A. Bulat, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "Toward fast and accurate human pose estimation via soft-gated skip connections," 2020.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[41] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *CoRR*, vol. abs/1406.2984, 2014.

[42] R. Kindermann and J. L. Snell, *Markov random fields and their applications*, vol. 1 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1980.

[43] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," *CoRR*, vol. abs/1411.4280, 2014.

[44] J. Bromley, W. Bentz, L. Bottou, I. Guyon, Y. Lechun, C. Moore, E. Sackinger, and R. Shah, "Signature verification using a siamese time delay neural network.," 1993.

[45] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[46] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1014–1021, 2009.

[47] M. Eichner and V. Ferrari, "Better appearance models for pictorial structures," 2008.

[48] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3041–3048, 2013.

[49] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR 2011*, pp. 1465–1472, 2011.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[51] Y. Huang, Y. Cheng, D. Chen, H. Lee, J. Ngiam, Q. V. Le, and Z. Chen, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *CoRR*, vol. abs/1811.06965, 2018.

[52] S. Zagoruyko and N. Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016.

[53] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[54] F. Chollet, "Keras." `https://github.com/fchollet/keras`, 2015.

[55] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," 2018.

[56] M. Själander, M. Jahre, G. Tufte, and N. Reissmann, "EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure," 2019.