NTNU

Norwegian University of
Science and Technology

Infant Body Part Tracking in Videos Using Deep Learning

Facilitating Early Detection of Cerebral Palsy

Kristian Aurlien

Daniel Groos

Master of Science in Computer Science

Submission date:	June 2018
Supervisor:	Herindrasana Ramampiaro, IDI
Co-supervisor:	Espen Ihlen, INM Lars Adde, IKOM

Norwegian University of Science and Technology
Department of Computer Science

Abstract

The breakthrough of Artificial Intelligence with the advent of Deep Learning has opened paths beyond what have earlier been explored. Within the medical domain, there are potentials to improve how problems are addressed and in the quality of the solutions. Computer-based methods have proven to facilitate early detection of cerebral palsy, which can make a difference by enabling treatment that can reduce the extent of disabilities in affected children. As these systems depend on motion patterns, information about the movement of infants must be collected.

In this thesis, we propose a method for tracking body parts of infants in video recordings. The developed framework addresses the task by identifying body parts frame by frame. In this way, the approach can be related to existing methods within the domain of Computer Vision, and more specifically the task of Human Pose Estimation. Taking advantage of recent progress in Deep Learning, the proposed Convolutional Neural Network outperforms existing techniques within the field, by localizing body parts of infants more precisely and at the same time operating at a speed of 120 frames per second. A large dataset of 140 700 annotated keypoints is constructed to facilitate this development. The proposed method has the potential of constituting an essential part of a system able to detect cerebral palsy at an age when the brain still has the ability to adapt.

Sammendrag

Med gjennombruddet innen kunstig intelligens ved fremveksten av dyp læring har det åpnet seg muligheter som aldri før har blitt utforsket. Innen medisin er det store potensialer for fornyelse, både i måten problemer er løst og kvaliteten på løsningene som finnes. Databaserte metoder har vist seg å fasilitere tidlig deteksjon av cerebral parese. Dette kan utgjøre en forskjell ved å muliggjøre behandling som kan redusere omfanget av funksjonsbegrensninger for barn som er rammet. Ettersom disse systemene benytter bevegelsesmønstre til deteksjon av cerebral parese er det nødvendig at informasjon om spedbarnets bevegelser kan hentes ut.

I denne masteroppgaven foreslår vi en metode for å følge kroppsdeler i videoer av spedbarn. Det utviklede rammeverket løser oppgaven ved å identifisere kroppsdeler i hvert bilde av videoen. På denne måten kan tilnærmingen til problemet relateres til eksisterende databaserte visuelle metoder og mer spesifikt Human Pose Estimation. Ved å dra nytte av fremskritt innen dyp læring foreslår vi et konvolusjonelt nevralt nettverk som utklasser eksisterende metoder ved å lokalisere spedbarns kroppsdeler mer nøyaktig, og som prosesserer 120 bilder i sekundet. For å oppnå dette har et datasett med 140 700 annoterte nøkkelpunkter blitt utviklet. Metoden kan bli en viktig del i et system som er i stand til å detektere cerebral parese i en alder der hjernen fremdeles har evnen til å tilpasse seg.

Preface

This thesis is submitted to the Norwegian University of Science and Technology upon completion of the five years Master of Science programme of Computer Science. The project is conducted as a collaboration between Kristian Aurlien and Daniel Groos.

In this project, we have addressed a problem faced by a multi-disciplinary research group consisting of clinicians and researchers at St. Olavs University Hospital and the Norwegian University of Science and Technology. Associate Professor Heri Ramampiaro at Department of Computer Science has been the main supervisor of the Master's project, with Associate Professor Espen Alexander F. Ihlen at Department of Neuromedicine and Movement Science and Dr. Lars Adde at Department of Clinical and Molecular Medicine as co-supervisors.

Acknowledgement

First of all, we want to express our sincere gratitude to our supervisor, Associate Professor Heri Ramampiaro, for giving us the opportunity to work on this project, and for his guidance, support and enthusiastic encouragement and insightful critiques during the various phases. He has also given us numerous opportunities to present and share our work, which has been of great inspirational and educational experience.

We are also genuinely grateful to Dr. Lars Adde and Associate Professor Espen Alexander F. Ihlen, our co-supervisors, for the essential expertise and close follow-up provided through all stages of the project. Together with Astrid Ustad, they also performed a substantial contribution in the data annotation work of this project. We thank you all for the many hours you put into this.

A special thanks for reviewing and proofreading our thesis goes to Harald Aurlien and all our supervisors, your feedback has been invaluable. Finally, we want to thank our girlfriends, friends, and family. You are our *raisons d'être*; without you by our side, all this work would not have been meaningful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Motion tracking is an essential component in many applications. In medical settings, movements might reveal information useful for clinical diagnosis. Diagnosis of cerebral palsy (CP) is a medical research area that can benefit from methods performing accurate tracking of infant movements.

CP is defined as a group of permanent disorders of the development of movement and posture that occur in the developing fetal or infant brain [3]. Existing techniques for diagnosis are complex and can not be conducted before 18 months of age. Movements indicative for the absence of CP involve a variable sequence of arm, leg, neck and trunk movements [4]. Keeping this as a starting point, Adde [5] hypothesizes that CP could be detected by documenting the absence of these patterns. As a result, motion tracking could be an important part of an automatic solution that would serve both patients and clinicians as a tool for CP detection.

First and foremost, such a system would make early detection of CP available to a higher number of people. Detection of CP might be performed already when the infants are as young as three months old when the brain is still plastic. Consequently, children with CP can receive more effective treatment. Second, clinicians can save time, and focus on areas requiring their competencies to a greater extent. Finally, because the technique is non-invasive, infants are spared for any unnecessary risks related to diagnosis, such as painful and potentially harmful anesthesia.

1.2 Problem Statement

In general, little emphasis has been made on adapting modern computer-based methods to important real-world problems. Powerful Artificial Intelligence (AI) techniques, including Deep Learning, have not been successfully applied to various mature disciplines demanding renewal.

When it comes to identifying disabilities in newborns through motion tracking of videos, existing techniques are both too imprecise and inefficient. In particular, current solutions for detecting movements depend on manual work and expensive computational resources. On the other hand, Deep Learning methods solve similar tasks more efficiently and at the same time with high precision. However, such methods currently only work well with grown-up individuals. This is mainly because of the lack of an appropriate dataset, but it might also be as a consequence of suitable models not being constructed yet.

By taking advantage of recent achievements in Computer Vision and Deep Learning, the primary goal with this project is to make body part tracking of infants available and accessible for medical researchers and clinicians to use in their daily practice.

Moreover, a solution for detecting CP through automatic movement analysis might inspire similar solutions in other medical domains. This could be achieved by proving that AI offers sufficiently explainable methods which could be integrated in medical applications, specifically by providing information from intermediate steps through a subsystem such as a body part tracker.

By proposing a method on the task of tracking body parts in videos of infants, one might also improve performance in other areas where Deep Learning could be applied.

To summarize, a successful method for tracking movements of newborns utilizing Deep Learning techniques could serve a purpose both when it comes to diagnosis of CP, medical applications in general, and other areas that might benefit from the developed methodology. Most important, the research could result in relevant infants living a life with less severe disabilities.

1.3 Research Questions

The main goal of the thesis is to develop a fully automated method for localizing and tracking body parts of infants in video recordings. As part of this, we propose a Deep Learning model that is able to perform the task better than existing state-of-the-art approaches within the domains of Computer Vision and Deep Learning. As a secondary goal, it is desired that the obtained method is able to perform the task efficiently without requiring use of costly and complex supercomputer systems. This is an important aspect considering the method is intended to be applied in a real-world medical system. With this in mind, this thesis aims at addressing the following research questions:

1. *Can a Deep Learning model solve the task of Infant Body Pose Estimation accurately?*

2. *Can this be done efficiently such that the model can be integrated in a system for medical diagnosis?*

1.4 Research Method

This project is based on a quantitative study where alternative solutions are evaluated based on their statistical performance on the relevant problem. Initially, relevant approaches are assessed to gather information about the state-of-the-art methods related to the problem. Based on the analysis of these methods, we hypothesize what can be improved with the given methods and use this as a guideline when proposing our method. Before evaluating the proposed method and comparing it with the state-of-the-art approaches, appropriate evaluation metrics are selected, and a dataset on which experiments are performed is acquired. Subsequently, experiments are carried out, and comparisons of the different methods can be obtained. Accordingly, observations and conclusions are extracted from the experiments.

1.5 Context

The project is part of a larger research initiative as a collaboration between St. Olavs University Hospital and the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The group consists of clinicians and researchers from the Clinic of Clinical Services and the Department of Pediatrics at St. Olavs Hospital, as well as from the Department of Clinical and Molecular Medicine and the Department of Neuromedicine and Movement Science at NTNU. The initiative is referred to as Computer-based Infant Movement Assessment (CIMA) and has the overall goal to improve the existing techniques for diagnosis of CP in premature infants. More specifically, there is a strive to obtain a system that can automatically detect CP solely based on a video recording of the relevant infant. Thus, it is desired that the diagnosis is performed based on the movement characteristics of the infant. At the same time, the system should provide meaningful insights to clinicians about the diagnosis and hence should facilitate the use of intelligent computer-based solutions in medical settings.

1.6 Contributions

As a first step towards obtaining a system that can detect CP automatically from a single video recording, it is necessary to ensure that information about the movement of the infant is correctly extracted from the video recording.

The main contribution of this thesis is to develop a method able to localize and track body parts of infants precisely across video frames. The developed methodology will constitute the first major building block in a more extensive system that aims to outperform current solutions for early prediction of CP. The body part tracker should offer improved performance, both in terms of precision and efficiency, compared to state-of-the-art approaches. In this way, the thesis will be a contribution to the medical domain as well as the Artificial Intelligence society.

As a basis for the development of a fully functioning body part tracker, a large dataset of annotated video frames is constructed. The dataset is based on a large collection of videos of infants at risk of developing CP. This facilitates the development of data-hungry computer-based methods in domains requiring data related to the pose of infants.

1.7 Thesis Outline

The thesis is structured as follows:

- **Chapter 1** explains the main motivation of the project, which specific research questions are addressed and in what way the research is conducted. We also consider the contributions of the project and the relation to the overall research initiative it is involved in.

- **Chapter 2** introduces theory essential for the project. This includes an introduction to the relevant medical domain as well the concepts of Computer Vision and Deep Learning.

- **Chapter 3** discusses state-of-the-art approaches related to the problem being addressed in this project.

- **Chapter 4** describes our methodology and the proposed method for localizing and tracking body parts in videos of infants.

- **Chapter 5** documents the results obtained in the research and how the proposed method performs compared to state-of-the-art approaches.

- **Chapter 6** comprises of the final discussion where results are evaluated and the applicability of the constructed method is assessed.

- **Chapter 7** contains the overall conclusion of the thesis and suggestions for future work.

Chapter 2

Theoretical Background

In this chapter, we introduce and give an overview of some of the theory providing the basis for the remaining chapters of the thesis. Initially, CP is characterized, and concepts related to early detection of CP is explained. Subsequently, the domains of Computer Vision and Deep Learning are presented, followed by an introduction to theory relevant for the problem being addressed in this project.

2.1 Cerebral Palsy

CP is the most frequent neurological movement disorder in children. With a prevalence of around 9%, premature infants have a particularly high risk of developing CP [6,7]. CP causes loss or impairment of motor function. Typical effects of CP include lack of movement control, speech difficulties, unnatural posture and inability to maintain balance.

In general, CP develops as a result of a brain injury or abnormal development of the immature brain [8, 9]. These injuries can in many cases be detected using Magnetic Resonance Imaging (MRI) or Computed Tomography scan (CT), which are some of the widely used methods for diagnosis of CP today. However, these methods are expensive, not available to everyone, and highly dependent on experts for image interpretation. Moreover, MRI and CT require the use of anesthesia, which involves particular risks for infants and young children.

The disturbances of the brain leading to CP are non-progressive, meaning that the damage or anomaly of the brain will not worsen over time. Nevertheless, the clinical consequences like pain and limitations of joint and muscle movement can become more severe over time. Because the plasticity of the brain is higher until two years of age, it is crucial to detect CP at an early stage such that appropriate therapy can be given when the brain still has a high ability to adapt [10]. Accordingly, with early treatment, a child might experience fewer difficulties growing up by for instance being able to walk without assistance.

2.2 Fidgety Movements

In order to initiate early treatment of infants with CP, the condition should be detected at an early age. However, the signs of CP are not usually present during the first years of life [11]. In spite of this, during the early development, a movement pattern useful for functional assessment of the young nervous system appears.

According to Prechtl [4], general movements are recognized as complex fluent and elegant movements that occur frequently and involve the whole body (arms, legs, neck, and trunk) in a variable sequence of movements. The general movements that typically appear between week 9 and 20 post-term are referred to as fidgety movements. These are in particular characterized as small movements of moderate speed in all directions. The absence of fidgety movements poses a strong indication of CP [12].

By utilizing the absence of fidgety movements as the basis to predict CP, a sensitivity of 98% and a specificity of 91% was shown in a study comprising 326 children [13]. From Equation 2.1, it can be inferred that the sensitivity measure defines the proportion of infants actually developing CP that were predicted to develop CP based on the absence of fidgety movements. This is often referred to as recall. On the other hand, specificity (Equation 2.2) denotes the degree to which the method is able to correctly predict that an infant does not develop CP.

$$Sensitivity = \frac{True\ positives}{True\ positives + False\ negatives} \tag{2.1}$$

$$Specificity = \frac{True\ negatives}{True\ negatives + False\ positives} \tag{2.2}$$

2.3 General Movement Assessment

Einspieler et al. [14] proposed a method to assess the general movements present during the course of development. More specifically, by General Movement Assessment (GMA) the movement pattern of an infant at 3 months of age is assessed in order to detect if the infant produces the fidgety movements that are associated with normal motor function.

In order to accurately identify whether fidgety movements are present or absent, clinicians are trained to recognize this movement pattern by observing video recordings of the general movements of various infants. This type of GMA is referred to as Gestalt Perception [15]. Gestalt Perception involves the evaluation of the complexity of general movements not by paying special attention to individual movements of the body parts but rather the overall movement repertoire of the body. Observers with the required training are able to efficiently recognize fidgety movements by Gestalt Perception. In this way, prediction of CP can be obtained accurately [10].

Although Gestalt Perception is a robust way of assessing general movements, there are some limitations to this approach. First, the technique highly depends on skilled personnel for observing the movements of the infant to come up with a reliable analysis. Such personnel might not be available in ordinary clinical practice. Second, assessment of general movements using Gestalt Perception is subjective. The analysis is therefore prone to be affected by external factors, such as fatigue and personal bias of the observer.

The challenges with the manual, qualitative approach of GMA makes it natural to explore approaches for detecting CP that can be conducted with minimal human interaction. For this purpose, it is natural to turn to computer-based methods. To be able to predict CP correctly based on computer software, we assume that relevant motion information must be extracted from the videos. Additionally, we hypothesize that a system can understand aspects of motion information related to the development of CP, such that automatic analysis can be conducted to perform accurate detection of CP.

In Section 3.1.1, approaches for performing computer-based GMA are discussed.

2.4 Computer Vision

In order to solve our movement tracking problem, we need systems that are able to reason about images. Such tasks are covered by the field called Computer Vision, which deals with automation of abilities of the human visual system [16]. The field thus covers a broad range of topics related to image tasks, ranging from low-level feature extraction to the understanding of concepts represented in such data. Examples of relevant problems which lean towards the low-level spectrum of this space include edge detection, noise removal, and image sharpening. In the other end of the spectrum, we find tasks like image classification, event detection and object localization [17]. While the primary goal in Computer Vision is to give computers the abilities of the human visual system, it has also given us a deeper understanding of how the human visual system actually works.

The low-level tasks of Computer Vision can often be solved by looking at the structure and qualities of the data itself. Edge detection, for instance, can be performed by searching for high levels of change in light intensity over a small spatial distance. Problems like image classification are on the other hand much harder to solve in this manner. The world contains a lot of complexity, and it seems reasonable to think that high-level reasoning about images from the real world requires some level of reasoning and intelligence.

Following this argument, recent improvements in Computer Vision have arguably been dominated by the research in Deep Learning, a sub-field of AI [18].

2.5 Deep Learning

In 2012, a team of researchers at the University of Toronto started a revolution in the field of Computer Vision, when they won the prestigious ImageNet competition on classifying objects in images using a deep neural network [19].

During the years that followed, this approach has proven to be very effective for machine learning tasks in general, and in Computer Vision in particular. Numerous research studies have shown that the neural network model can be used as a general framework to solve a great number of tasks, given enough training data. In this section, we will therefore provide a brief introduction to the concepts behind neural networks and Deep Learning.

2.5.1 Neural Networks

As already indicated, one of the most fundamental building blocks of Deep Learning is the neural network. A neural network is a computational graph utilized to discover patterns in the data presented to the model at the input layer, such that appropriate outputs can be produced at the output layer. The intermediate processing between the input and output layers of a neural network happens in the hidden layers.

The architecture of a neural network may be designed in a variety of different ways to suit our needs regarding the simplicity or complexity of the task that we are dealing with. Neural networks thus offer a framework suitable for almost any task when information can be represented by numerical values. Typical design choices involve the number of hidden layers in the network, the number of units in each layer, and the type of mathematical operations applied at each stage of the graph.

The design of the network architecture is essential to obtain high performing models. More hidden layers and more units in each layer lead to more complex models, which can represent more advanced functions. Contradictory, the high number of parameters in such complex models can also lead to degradation of accuracy because they are harder to optimize. Models might also become slower as a result of more parameters involved in the computations within a neural network.

A simple neural network producing a single output value from three input values through two hidden layers of four units is illustrated in Figure 2.1.

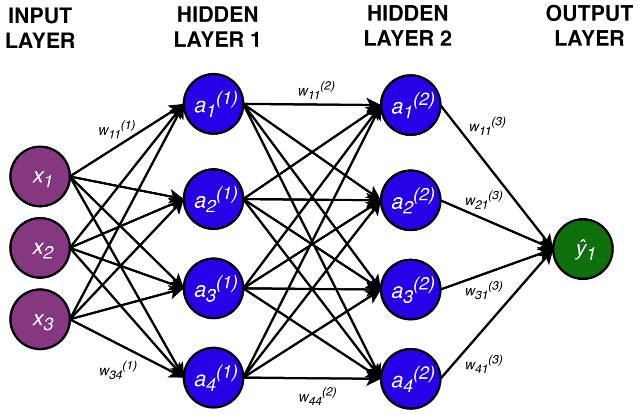


Figure 2.1: A simple fully connected neural network

Learning

To learn which aspects of the inputted data the model should emphasize to produce the desired output, neural networks incorporate a learning mechanism, which can be considered an optimization process. The model is presented with a set of examples, which are used to guide the network on how the respective task should be solved. In supervised learning, each example consists of a feature vector x describing the input values and a target vector y defining the optimal solution given the features of that example.

For a neural network to be able to generalize upon the examples shown to the network, the examples should be of such variety that they represent most of the diversity of situations within the given task. As a result, many examples are often required to achieve proper learning in neural networks.

The learned patterns of neural networks are embedded in connections between units of subsequent layers. In particular, the connections are assigned values referred to as weights (W) which can specify the significance of different connections in the network. The weights are updated during the learning process by utilizing the mechanism of backpropagation.

Backpropagation aims to minimize the error of the model when predicting an output vector and correspondingly alters the weights of the model to reduce this loss in future occasions. For each iteration, the weights are updated a step towards the opposite direction of the gradient of the loss with respect to the weights. In this way, as learning proceeds more optimal weights are obtained and the error is decreased. The step size that is used for updating weights, is usually expressed as the *learning rate*. This specific strategy of optimizing the weights of a neural network is most commonly referred to as Stochastic Gradient Descent (SGD). Further details on how learning is performed in neural networks using SGD, can be found in Chapter 8.3.1 of Goodfellow et al. [16].

Rather than updating weights solely based on the current gradient, the SGD algorithm is extended to take into account both the most recent information as well as the gradients of the previous iterations. This technique is referred to as momentum [16]. Momentum facilitates steady learning by consistently providing weight updates in a more optimal direction. Because of this, momentum tends to accelerate training.

In the process of optimizing the weights of a neural network, it is important that the loss function utilized for this purpose is selected according to the type of output expected from the model. This is crucial as the objective of the learning procedure is to obtain weights that produce desired outputs given any input data.

Forward Pass

The forward pass of a neural network refers to the process of estimating appropriate outputs based on a set of input values. More specifically, the input values x are presented to the network at the input layer and goes through a chain of operations before the output estimate $\hat{y}$ can be obtained at the output layer of the network.

For each layer in the neural network, the forward pass consists of two different operations. First, weights are utilized to compute how each value in the previous layer contributes to the inputted value of the units of the subsequent layer. In the case of fully connected layers (Figure 2.1), which have connections between all neurons of two subsequent layers, the scalar input for each unit will depend on the activation value of all units in the layer earlier in the network. Accordingly, the weighted input of a single unit constitutes a sum which is based on all activation values of the preceding layer together with the associated weights. The weighted input operation can be compactly expressed as the matrix-vector product in Equation 2.3, when we are treating $a^{(l-1)}$ as a row vector defining all activation values of layer *l-1* and $W^{(1)}$ as the weight matrix of layer *l*.

$$z^{(l)} = a^{(l-1)}W^{(1)} \tag{2.3}$$

Following the computation of weighted inputs, the second operation of the corresponding layer is performed. This refers to the process of calculating activation values, and is performed by applying an activation function $g(\cdot)$ on the weighted inputs $z^{(l)}$. The main purpose of the activation function is to introduce non-linearity to the neural network model. By performing non-linear functions during the forward pass, a neural network becomes capable of learning complex non-linear patterns in the data. Partly, this is a reason why neural networks are able to solve highly difficult tasks. In spite of its simplicity, a very powerful activation function is the Rectified Linear Unit (ReLU) as defined in Equation 2.4. In particular, ReLU suits well with neural networks that are deep by means of consisting of many hidden layers. This is often beneficial for image-related tasks as such problems often requires the use of very deep neural networks.

$$g(z^{(l)}) = max(0, z^{(l)}) \tag{2.4}$$

2.5.2 Convolutional Neural Networks

For tasks involving images, a particular type of neural networks called *Convolutional Neural Networks* (ConvNets) has proven useful.

ConvNets employ the mathematical principle of convolution to have local connections between units of consecutive layers. Consequently, each unit in a convolutional layer is only concerned with a subset of the information available, determined by the specific region of the image from which the unit receives inputs. This region is more commonly known as the receptive field of the unit. The concept of local connectivity results in each unit acting as a feature detector for a particular part of the image. The values of the weights of the local connections determine the kind of feature a unit responds to.

A second important characteristic of convolutional layers is weight sharing. In the case of convolution, weight sharing refers to the principle that the same set of weights is used to calculate the response for each of the units in the same feature map, although they receive inputs from different regions of the image. In other words, all units are looking for the same type of feature across the image. Moreover, because a single set of weights, more frequently referred to as a kernel or filter, can only detect a specific type of pattern, it is usually valuable to apply several different sets of weights in a single convolutional layer.

The use of several different kernels in a convolutional layer results in producing several feature maps, where each feature map is a pool of units associated with applying one specific kernel on the image. In a convolutional layer, the different feature maps are stacked upon each other producing an additional dimension. Figure 2.2 displays how three feature maps of four units are obtained from an input image, when applying convolution with three kernels (W) each of size 2×2. Utilizing a kernel with height and width of 2 is often referred to as 2×2 convolution.

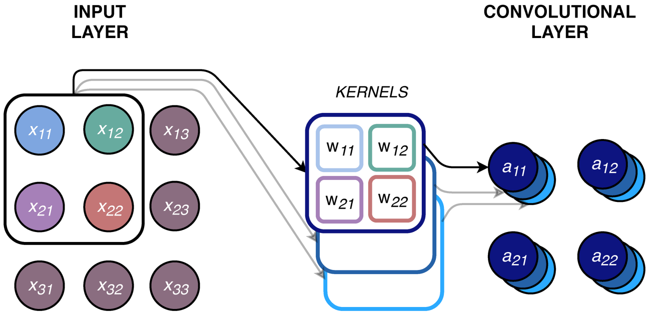


Figure 2.2: A simple convolutional layer. The illustration highlights the image region that corresponds to the receptive field of the upper-left units of the convolutional layer.

ConvNets applied to actual tasks usually require both complexity and consideration in terms of network architecture. First of all, an additional type of layer, called downsampling or pooling layers, is often applied for reducing the resolution of the feature maps of convolutional layers. There are different types of downsampling operations, such as max pooling and average pooling, which emphasize different characteristics of the feature maps. Downsampling might be helpful for several reasons, including the ability to store information in a less expensive format to reduce computation. Downsampling also introduce some form of translation invariance to a model. Translation invariance can lead to improved statistical efficiency of the ConvNets [16].

As with traditional neural networks, the depth of ConvNets affect the expressiveness of the network in terms of the complexity of patterns that can be extracted. More concisely, utilizing an architecture consisting of several convolutional layers enables layers later in the network to learn more complex features, which in turn are computed from simpler features learned by convolutional layers earlier in the network.

When dealing with images, we can consider this as if the initial convolutional layers detect simple features such as lines of specific directions, while the convolutional layers deeper down in the network are more concerned with identifying complete objects composed of many of these simple features.

There exist several types of convolutional layers. Recently, *dilated convolution* has proven promising [20]. Dilation introduces leaps in the input maps, which enables an increase in the receptive field of a unit without increasing the number of parameters. Figure 2.3 illustrates dilated convolution with a *dilation rate* of 2. This increases the receptive field of each unit from 2×2 to 3×3 compared to the original 2×2 convolution corresponding with a dilation rate of 1 (Figure 2.2). A benefit with dilated convolutions is thus the flexibility to design what should be the size of the receptive fields simply by increasing or decreasing the dilation rate of the corresponding layer. Consequently, with dilated convolutions a neural network might not require as many layers because each unit in a layer is capable of analyzing a larger portion of the image.

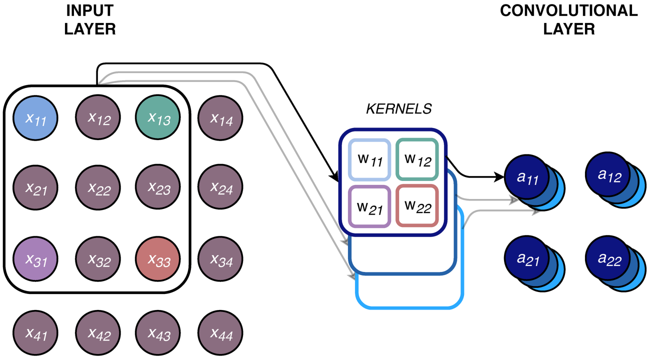


Figure 2.3: Dilated convolution performed with a dilation rate of 2.

Another alternative convolution operation employs the principle of *transposed convolution*, reversing the operation of convolution by obtaining a set of input feature maps when provided with the output. To increase the resolution of feature maps, transposed convolution can be applied. This is beneficial in many image-related tasks interested in obtaining high-resolution output from low-resolution intermediate feature representations.

To summarize, ConvNets are particularly suitable for solving tasks related to images. They are intuitive by having the benefit of considering spatial relationships present in images. At the same time, thanks to weight sharing and local connectivity, convolutional layers usually require fewer distinct weights than fully connected layers. Accordingly, the optimization process is simplified by having less parameters to tune during training. Consequently, there is also a low cost of introducing deeper ConvNets.

2.5.3 Transfer Learning

When training Deep Learning models, a large dataset is usually needed to achieve good results. In many real-world problems, collecting such a dataset is costly and time demanding. Transfer learning aims to solve this problem by applying a model trained on one problem on another, related problem. In this way, we can take advantage of large existing datasets, and apply models trained on these on other similar tasks.

The reuse of models can be performed in multiple ways. First, if someone already has constructed and released a model trained on a dataset similar to your own, it might be possible to directly apply the model and trained weights on your problem. Such models are often referred to as *pre-trained* models.

Second, it is possible to use a pre-trained model as a pre-processing step, and use parts of its internal layers as basis for a newly designed model. In this case, the pre-trained model is called a *feature extractor*. This approach takes advantage of the fact that deep neural networks tend to learn generic features, which can be useful across different domains [21]. For image-related tasks the feature extractor is commonly the convolutional part of a classifier network trained on the large ImageNet dataset containing several millions of labelled images [22]. During training of the new model, the weights of the feature extractor layers remains constant, while the rest of the network is updated by the learning process.

Finally, the weights of a pre-trained network can be used as a starting point when applying the same network to a related task. This differs from the feature extractor approach by all layers being trained, instead of keeping them fixed. However, fine-tuning can also be performed with the feature extractor approach. In practice, this is usually performed by having a separate step in the learning process where a lower learning rate is utilized after completing training of the other parts of the network.

2.6 Image Segmentation

One way to solve the problem of localizing body parts, is to assign a class to each pixel of an image. In this case, the class should ideally represent the body part the pixel is part of. By considering the classes of nearby pixels, we can gain an intuition of where different body parts exist in an image. This approach is called Image Segmentation, and is a sub field of Computer Vision. Specifically, Image Segmentation is concerned with the task of dividing an image into coherent segments.

Image Segmentation has proven useful in a wide range of situations, from medical imaging concerned with localizing the boundaries of a tumor among surrounding healthy tissue in an MRI-scan to automated vehicles where localization of objects plays an important part.

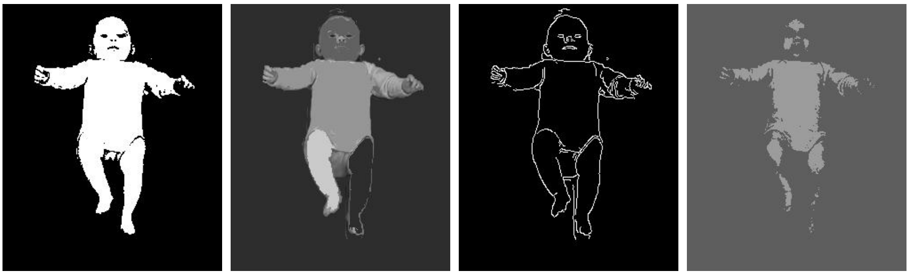
When discussing Image Segmentation, we often distinguish between *Semantic Segmentation* or *Instance Segmentation*. While the former can be described as the task of classifying each pixel in an image, the latter in addition is interested in separating between instances of the same class.

In the remaining part of this section, we will discuss approaches that have traditionally been applied to perform Image Segmentation of an image. Subsequently, in Section 2.7 an approach related to Image Segmentation that relates to our problem to an higher extent is introduced.

2.6.1 Traditional Approaches

One of the simplest approaches to Image Segmentation is called threshold-based segmentation (Figure 2.4a). Assuming that the background and foreground of an image differs in their pixel intensity values, one can use a threshold value to divide these two segments. Being dependent on this assumption, such methods will fail if for example the contrast between the foreground and background is too small, or if parts of the foreground has the same color and intensity as the background. As the pixel intensity values are dependent on lighting conditions and the content of the image, the threshold value has to be fine-tuned for each image. Usually, this is done manually or using some heuristic [23]. To introduce some more complexity to the method, an alternative would be to utilize several thresholds at the same time.

Another approach, which in addition takes into account some spatial information about the pixels in an image, is region growing [24]. In a bottom-up manner this can be done using a seed pixel as a starting point for a segment, and gradually grow the segment by adding nearby pixels which are close in intensity value. Region-based segmentation can also be done in a top-down manner, by splitting the image into smaller segments until some similarity measure within each segment is met. Additionally, splitting and merging can be alternated, such as the proposed strategy of Horowitz et al. [25]. Although the region-based methods are not only based on pixel intensity values, the issues of low contrasts seen in threshold-based methods are still present. Correspondingly, a region-based segmentation algorithm is only capable of separating different segments from each other if there are clear boundaries in the image (Figure 2.4b).



(a) Thresholding (b) Region-based (c) Edge detection (d) k-means

Figure 2.4: Traditional image segmentation techniques

Segmentation can also be done by first detecting edges in an image, and using these edges to identify the connected areas. An edge is typically seen as a quick change in color or intensity value over a small amount of pixels, in some direction. Similar to the already mentioned approaches, this also requires good contrasts in the image in order to give good results. The illustration of Figure 2.4c shows that edge detection-based segmentation to some extent manages to distinguish body parts, but is overly sensitive to noise and can result in over-segmenting the image.

Clustering-based approaches uses a notion of closeness to try to gather similar elements into groups or clusters. A common algorithm following this approach is called k-means [26], which tries to split a set into k separate clusters while minimizing the distance measure within each cluster. The distance measure can be defined in various ways, and can typically take into consideration the intensity value, the distance in space, or both. Similarly to the aforementioned approaches, clustering-based approaches are dependent on contrast, and as displayed by Figure 2.4d this might lead to less coherent segments being predicted.

In summary, these traditional methods for performing Image Segmentation all have weaknesses which exclude them for our purpose. In particular, they all depend heavily on light-intensity or color-intensity of image pixels to extract segments from an image. This makes the methods highly dependent on stable and good lighting conditions, and they are at the same time less robust in case of changes in skin color, clothing or background. In addition, they make no use of prior knowledge about the objects present in the images.

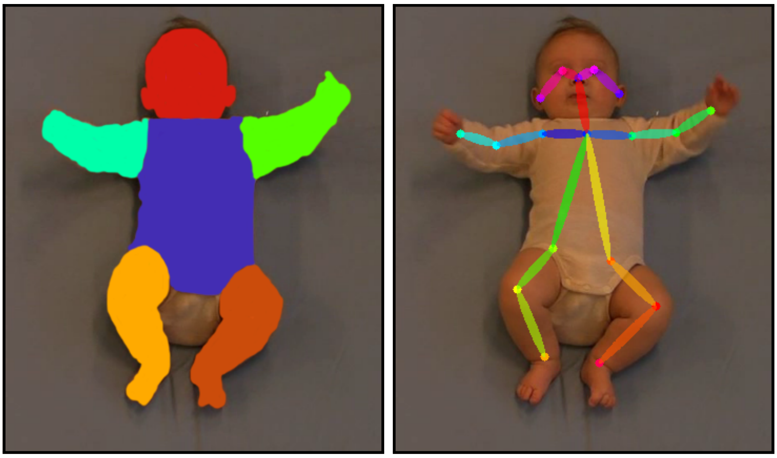
2.6.2 Deep Learning-Based Approaches

While the traditional methods can prove useful in simpler tasks like background removal, they often fail to solve more complex problems where high-level understanding and reasoning are needed. In such cases, Image Segmentation methods based on Deep Learning have proven highly effective. In particular, by learning features from a large set of training data, Deep Learning models are able to extract general knowledge of concepts within images. This can be utilized to perform accurate segmentation of images never previously presented to the model.

2.7 Human Pose Estimation

Human Pose Estimation can be defined as the task of estimating the configuration of the human pose from an image [27]. This task has gained substantial interest from the Computer Vision community in the later years, playing an important role in applications ranging from automated marker-less Motion Capture [28] to Activity Recognition [29].

This task can be considered a specialized version of Image Segmentation and Object Localization. As the goal of this project is to obtain a system able to localize and track body parts of an infant, we do not require detailed information on pixel-level provided by Image Segmentation methods. However, Human Pose Estimation is only concerned with extracting information related to the configuration of the pose of the human of interest. Therefore, as illustrated by Figure 2.5, Human Pose Estimation can be utilized to directly estimate the locations of body parts without requiring a more extensive segmentation step to be performed.



(a) Image Segmentation (b) Human Pose Estimation

Figure 2.5: Image Segmentation and Human Pose Estimation as two different approaches to localize body parts in an image

Chapter 3

State of the Art

In the previous chapter, general ideas of GMA and Computer Vision were introduced, including how these fields relate to our problem. However, we did not go into detail on how these concepts are applied to solve the tasks we are concerned with.

The following chapter shows how these ideas are incorporated in computer-based methods, covering the state of the art in relevant fields. Section 3.1 presents existing methods for solving GMA and Image Segmentation, and what challenges these approaches have encountered. Subsequently, Section 3.2 gives an overview of modern methods in Computer Vision which relate to the way we are addressing the problem. This includes using Deep Learning to perform Feature Extraction and Human Pose Estimation.

3.1 Alternative Approaches

3.1.1 Computer-Based GMA

As elaborated in Section 2.3, computer-based methods for GMA are desired to overcome challenges with conventional approaches for GMA. The main limitations a computer-based system strives to overcome are the dependency to trained personnel for performing the analysis as well as the subjective nature of the manual strategy to detect CP.

During recent years, several computer-based methods have been utilized to analyze the infant's movement in a non-obtrusive way based solely on video recordings. The proposed approaches differ to a large extent in terms of complexity and correspondingly solve the task of predicting CP with varying degrees of success.

Within the CIMA initiative, there are in particular three computer-based approaches for detection of CP that have been developed and thoroughly tested. These approaches can be characterized by a two-stage process. The first step involves extracting motion information from the videos. The motion information should essentially contain factors relevant for describing the fidgety movements of the infant. This information is represented as a feature vector. The second stage of the process is concerned with the procedure of predicting CP based on the motion information extracted in the previous step.

Frame Differencing

The computer-based approach developed by the CIMA group that first appeared to be promising in analyzing video recordings to predict CP, utilizes the technique of Frame Differencing [30, 31]. Specifically, the method of Adde et al. from 2009 represents information about the motion of the infant through motion images expressing changes in pixel intensities of subsequent frames. From the motion images the centroid of motion is extracted and passed as input to the prediction stage.

Based on the standard deviation of the centroid of motion, a threshold is used to perform prediction of CP. The findings show that a low standard deviation is likely to indicate fidgety movements being present. This is supported by the definition of fidgety movements as frequent movements involving the whole body and thus can be associated with a stable centroid of motion.

While being simple and computationally efficient, this method yields a sensitivity of 85% and a specificity of 71% [31]. However, there are some drawbacks regarding the robustness of the approach. For example, the motion image is sensitive to changes in lighting conditions, differences in clothing and skin color and motions not related to the movements of the infant. As a consequence, the method is not easily applicable in clinical settings.

Optical Flow

The method utilizing Frame Differencing does not attempt to extract information representing the movements of different body parts. With the advent of an Optical Flow-based approach in 2012, motion trajectories are extracted from the videos [32]. Optical Flow is used to extract measures describing the speed and direction of objects in consecutive frames, which are further taken into account to produce the motion trajectories. The final feature vectors are based on a histogram representation of the time distances between consecutive maxima in the trajectories.

These features can be utilized by a Support Vector Machine to predict CP. Stahl et al. [32] illustrated improved accuracy of computer-based methods to detect CP, with sensitivity and specificity of 85.3% and 95.5% respectively. The results provided by the optical flow-based approach show that normal fidgety movements display less variety in the time distance measure. This can be associated with circular motion patterns often present in fidgety movements.

A drawback with this approach is that Optical Flow is computationally expensive and highly depends on the availability of supercomputers. Additionally, there are no ways to ensure that tracked image points reflect the movement of only a specific body part. This leads to occlusion and drifting in challenging cases.

The third and most recent approach differs from the previous in the way the motion trajectories are utilized to extract features. In particular, Rahmati et al. [33] overcomes the shortcomings of occlusion and drifting by dividing trajectories into groups representing distinct body parts explicitly using manual work [34,35]. From this grouping, a single trajectory is chosen to represent each body part. By applying the Fast Fourier Transform, frequency components are extracted from the motion trajectories which further gives rise to the final feature vectors.

With this approach, detection of CP is performed by applying Partial Least Squares Regression. The method yields sensitivity and specificity values of 86% and 92% [33]. Besides suffering from the previously mentioned expensive computational cost of Optical Flow, an obvious limitation of the method is the dependence of human annotation for proper segmentation and tracking of body part movements.

Random Ferns

Neither the application of Frame Differencing nor Optical Flow provides accurate movement information in a fully-automatic manner. As a result, alternative approaches might be considered necessary to extract infant movement information.

In 2015, Hesse et al. [36, 37] proposed a system able to localize infant body parts using a machine learning algorithm called Random Ferns. Acting completely automatically, this method is promising in obtaining the locations of different body parts. However, the technique uses computer generated depth images as input. Consequently, it can not be easily adapted to normal video recordings, where there is no explicit information about the depth in an image.

Considering currently available approaches for extracting movement information of infants using video recordings, it becomes evident that there is need for a method that is able to perform the task both accurately and efficiently.

3.1.2 Image Segmentation Techniques

An alternative way to extract motion information is through Image Segmentation. Like many other sub fields of Computer Vision, Image Segmentation has been strongly influenced by Deep Learning. In Section 2.6.1, we described a set of traditional methods and their shortcomings. However, recent development in the field has improved performance to a large degree. Hence, combined with the relevance of Image Segmentation in localizing body parts, state-of-the-art methods for performing Image Segmentation was one of our focus areas while reviewing literature. Although these methods could suffice for solving our problem, as explained in Section 2.6, they provide a less direct path towards the solution compared to Human Pose Estimation. Still, many improvements in Human Pose Estimation have been inspired and influenced by the development of Deep Learning-based Image Segmentation. As a result, we provide a brief overview of the development of the field without going in too much detail.

The task of Image Segmentation received its breakthrough in 2015, with the introduction of Fully Convolutional Networks [38]. Long et al. showed how convolutional networks could be trained on this task in an end-to-end manner, with results exceeding current state-of-the-art methods. Importantly, they demonstrated upsampling (decoding) of features from low resolution to high resolution as a powerful building block.

In 2016, Badrinarayanan et al. [39] published the SegNet architecture, which further developed the idea of upsampling. Realizing that prior architectures used the majority of their parameters on the feature extraction (encoding) part, they proposed a more balanced structure with fewer layers of feature extraction, but where each layer had a corresponding decoding layer. Compared to Long et al. [38], SegNet reduces the number of parameters from 134 million to 15 million.

Another recent approach, Mask R-CNN [40] combines ideas from Object Localization and Instance Segmentation. Mask R-CNN is based on the powerful Faster R-CNN [41] architecture to predict bounding boxes for objects of interest, and a Fully Convolutional Network to provide pixel-wise segmentation of these areas. Altogether, Mask R-CNN displays state-of-the-art results on Instance Segmentation. Additionally, He et al. [40] demonstrated the flexibility of the model, by adopting it to the task of Human Pose Estimation. This was achieved by assigning one segmentation class per body part, and predicting one-hot segmentation masks representing body part locations.

3.2 Related Work

3.2.1 Deep Feature Extraction

As illustrated by Section 3.1.2, ConvNets provide state-of-the-art methods for solving tasks within Computer Vision. Overall, what makes this approach especially suitable for image related tasks is the ability to extract deep hierarchies of useful features without the need for human intervention. Additionally, compared to conventional Computer Vision techniques, Deep Learning enables generalization beyond the examples presented to the neural network. In this way, complex patterns particularly relevant to a specific task can be learned, and thereafter utilized to obtain solutions to new cases.

In Section 2.5.3, we introduced the idea of a feature extractor as a ConvNet able to perform high-quality feature extraction by being trained on the massive ImageNet dataset [22]. Feature extractors are beneficial in tasks that require generic knowledge about the content of an image, such as Image Segmentation and Human Pose Estimation. Accordingly, feature extractors are setting the bar for how accurately image related tasks can be solved.

Although there is an immense amount of different Deep Learning models aiming to provide superior feature extraction, some network architectures have made significant contributions to the field. Progress has been illustrated by yearly improvements on the Image Classification challenge of the ImageNet competition [42]. Importantly, the trained weights of these feature extractors have been made publicly available. As a result, they have been used as building blocks in many state-of-the-art methods within Computer Vision tasks.

AlexNet, proposed by Krizhevsky et al. [19] in 2012, was the first ConvNet to win the ImageNet competition. The architecture consists of five convolutional layers and three fully connected layers. Although impressive when first released, AlexNet has later been surpassed by other models.

The ConvNets that have gained more recent popularity are characterized by deeper architectures and smaller kernels. Among these are VGG [43], a set of deep ConvNets developed by the Visual Geometry Group (VGG) at the University of Oxford. The most widely used VGG models include VGG-16 and VGG-19, which vary in the number of convolutional layers, 13 and 16 respectively. The main drawbacks with respect to VGG models are memory consumption and computational cost.

As a result, a 22 layer deep GoogLeNet [44] was introduced taking advantage of inception modules. The inception modules enable the use of kernels of different sizes in a single layer, such that convolutions of different types can be computed in parallel. In order to avoid further challenges in terms of computation and memory, 1×1 convolutions are utilized resulting in a decrease in the number of feature maps. This reduces both the number of parameters and number of operations, while providing slightly better accuracy than VGG (Table 3.1).

Another popular feature extractor, invented by Microsoft in 2015, is a very deep neural network called ResNet [45]. Very deep ConvNets had proven difficult to train because of a degradation problem in accuracy in networks consisting of very many layers. ResNet addresses this issue by utilizing residual blocks where identity shortcut connections are responsible of forwarding the information unaltered through the network. With a classification accuracy of 96.4% on ImageNet, the 152 layer deep ResNet clearly reflects that very deep neural networks can be used in the domain of Computer Vision with improved classification performance.

In 2017, Howard et al. [46] proposed a ConvNet as a response to existing networks being both slow and resource demanding. The MobileNet architecture should facilitate the use of accurate Deep Learning models in applications such as robotics or self-driving cars requiring low latency in spite of computational limitations. By utilizing depthwise separable convolutions, the network has no more than four million parameters although being 88 layers deep. Nevertheless, as illustrated in Table 3.1, the lightweight architecture of MobileNet comes at the cost of decreased classification accuracy.

The aforementioned feature extractors have proven to be either very accurate or less accurate but efficient. However, none have succeeded in fulfilling both criteria. With DenseNet, Huang et al. [2] propose an architecture that reuses features by connecting each layer to all subsequent layers in the corresponding part of the network. In this way, each layer of the network is able to learn more robust representations. At the same time, this yields efficient propagation in the network as well as low memory consumption. The DenseNet architecture illustrated high performance on the ImageNet classification challenge with a small pool of parameters compared to the great depth of the network.

The availability of high-quality feature extractors has accelerated the development of methods within the different tasks related to Computer Vision, including Human Pose Estimation. However, feature extraction in itself is not sufficient to solve such complex tasks. In the upcoming section, we introduce state-of-the-art approaches providing solutions to Human Pose Estimation using Deep Learning.

Table 3.1: Performance on the ImageNet classification challenge

Name	Year	Top-5 Accuracy	Depth	# Parameters
AlexNet [19]	2012	84.7%	8	62 300 000
VGG [43]	2014	92.7%	19	143 667 240
GoogLeNet [44]	2014	93.3%	22	11 193 984
ResNet [45]	2015	96.4%	152	60 344 232
MobileNet [46]	2017	89.5%	88	4 253 864
DenseNet [2]	2018	93.3%	121	8 062 504

3.2.2 Approaches for Human Pose Estimation

Like Image Segmentation, Human Pose Estimation deals with localizing objects in images. Accordingly, the development of Human Pose Estimation has in many ways been influenced by the models described in Section 3.1.2.

Early approaches for Deep Learning-based Human Pose Estimation directly predicted Cartesian coordinates of body parts [47]. However, the segmentation model of Long et al. [38] inspired more recent models to use spatial representations as output. In 2016, Wei et al. [48] started this trend for Human Pose Estimation models, which predicts one 2D *confidence map* per body part. Moreover, they proposed a multi-step network, in which each step uses a combination of the input image and the output from the previous step. In this way, intermediate supervision is performed to produce more accurate predictions later in the network.

Following along the same lines of intermediate supervision and confidence maps, Newell et al. [49] propose the Stacked Hourglass Network. The model uses subsequent downsampling and upsampling procedures in each step of the model, which facilitates prediction across multiple scales of the image. Downsampling is performed by pooling in multiple rounds, which produces intermediate representations of low resolution. Next an upsampling procedure is performed to ensure the output of each step has the same resolution as the input.

Yang et al. [50] builds upon the Stacked Hourglass structure, but modify the internal modules with residual connections, aiming to further facilitate learning from input features of different scales. This is motivated by observations showing that different camera angles and changes in scale occurring in natural images can be a challenge for the aforementioned architectures.

CMU-Pose [1] is a model from the team behind the Convolutional Pose Machine [48], in which Cao et al. add to their previous work some important modifications. First, the pre-trained classification network VGG-19 is used as a feature extractor, which gives a good initialization of the model and reduces training time. Second, they introduce a separate neural network branch for predicting body part connections. These connections are represented using vector fields referred to as *part affinity fields*. Intuitively, these vectors guide the prediction of reasonable postures of humans, by explicitly representing connections in addition to the body part keypoints. Moreover, the vector fields provide the model with the ability to separate keypoints from different people in an image, using a greedy bipartite matching algorithm. More relevant to our project, they also have a focus on inference speed, reporting 8.8 frames per second on a NVIDIA GeForce GTX-1080 GPU.

While the focus of Human Pose Estimation has usually been to find locations of body parts in a 2D-plane, some applications would benefit from also having the depth-dimension taken into account. A recent approach for solving this, released in 2018 by Güler et al. [51] is called DensePose. In addition to provide a 3D annotated dataset based on COCO [52], they conduct experiments combining ideas from aforementioned approaches [40, 48–50].

None of the models seen so far have taken advantage of *temporal* information, i.e. how we can better track body parts in video over time. Recent research projects by Insafutdinov et al. [53] and Iqbal et al. [54] suggest two different methods, ArtTrack and PoseTrack respectively, where such temporal information is also taken into consideration. However, their focus is on improving accuracy of crowded scenes, which is less relevant to our problem.

Chapter 4

CIMA-Pose

As presented in Chapter 3, problems related to the task we are dealing with have been addressed in various ways. Based on the research of existing approaches, we decided to utilize Deep Learning in combination with Human Pose Estimation for solving our problem. More specifically, we propose a model capable of predicting the location of body parts in an image. By applying the model to all frames of a video, a time series of body part locations is obtained. This provides a solution to the problem of Infant Movement Tracking.

In the upcoming chapter, we will go into more detail about our activities, experiments and proposed solution of this project. Initially, we describe the process of obtaining a dataset containing images of infants, an important prerequisite for the developed method. Building upon this, in Section 4.3, baselines for the performance of state-of-the-art approaches on the infant version of Human Pose Estimation (Infant Body Pose Estimation) are constructed. Section 4.4 provides a set of ideas aiming to facilitate the creation of a lightweight ConvNet. Subsequently, in Section 4.5, the proposed CIMA-Pose model is described. In the last part of the chapter, the optimization process of the models is described more thoroughly. This also includes an introduction to which evaluation metrics have been used and how the body parts are tracked through consecutive frames.

4.1 Keypoints

As specified in Section 2.7, Human Pose Estimation represents the pose of the human body using a set of keypoints. The selection of keypoints differs between projects, but is an important design decision to be made. In our project, the goal is to track the movements of body parts needed to predict CP. Previous work has shown that the movements of arms, legs, neck, and trunk are sufficient to accurately predict CP [5]. Based on this, and in agreement with the CIMA research group, we decided to focus on the seven keypoints displayed in Figure 4.1, with the following definitions:

1. **Nose (N)**: center of the nose tip
2. **Upper chest (UC)**: midway between the left and right shoulder
3. **Right wrist (RW)**: center of the right wrist joint
4. **Left wrist (LW)**: center of the left wrist joint
5. **Hip center (HC)**: midpoint between the left and right part of the iliac crest
6. **Right ankle (RA)**: center of the right ankle joint
7. **Left ankle (LA)**: center of the left ankle joint

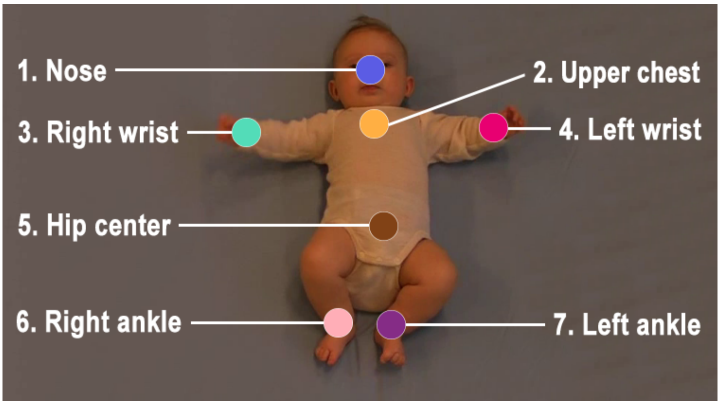


Figure 4.1: Definition of keypoints

Upper chest and *Hip center* are decided to represent the trunk for several reasons. Compared to a single trunk point, we find that these can be more clearly specified. This reduces ambiguity for human annotators which eventually might facilitate more robust predictions by the Deep Learning model. Second, this enables representing the movement of the body to more detail given that the alignment of the body might differ within the different poses. Finally, by representing the trunk with two keypoints a natural stick figure visualization can be obtained.

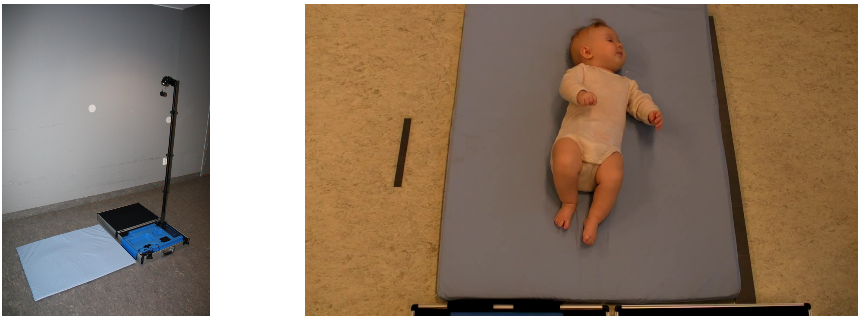
4.2 CIMA Keypoints Dataset

In the area of Human Pose Estimation, there are multiple benchmark datasets [52,55–57]. In recent years, the Common Objects in Context (COCO) Dataset [52] has been widely used. COCO contains more than 200 000 annotated images of one or more people in everyday situations. Our preliminary studies showed that while existing Human Pose Estimation models perform well on general data, they under-perform on pictures of infants. The reason for this might be the differences in body shape, differences in poses, or the fact that the infants we are concerned with are laying on their back. From this and based on the fact that the method is constructed in a supervised learning fashion, we argue that an annotated dataset of infants is necessary to make a stable body tracker that solves our problem.

This reasoning motivated the creation of the CIMA Keypoints Dataset, which consists of 20 100 annotated images of infants, in total 140 700 keypoints. In this section, we go into details on how we performed the data selection, and subsequently the annotation process is described.

4.2.1 Acquisition of Data

The research group at St. Olavs Hospital and NTNU has over many years, with collaborators, collected video recordings of infants. The recording process has been standardized using a self-developed set-up, consisting of a mattress, video camera, camera stand and a suitcase. The set-up is shown in action in Figure 4.2. The mattress size, 70 × 90 centimeters, is used as an estimate to measure distances in the recordings. More specifically, the height of the video frames can be roughly approximated to the height of the mattress (90 centimeters), as indicated in Figure 4.2b. The videos are recorded in clinical practice, of premature infants in the risk group of CP. The subjects are between 10 to 15 weeks post-term, wearing diapers or a single colored bodysuit.



(a) Camera stand and mattress

(b) Camera view

Figure 4.2: Standardized set-up of video recordings

To facilitate making our model as general as possible, videos originating from multiple sites are taken into account while constructing the dataset. 67 videos were selected from each site, using stratified sampling to retain each site's prevalence of CP. More details about the original selection of videos are summarized in Table 4.1. Among the selected videos, 80% from each site is used for training and validation, while the remaining 20% is kept separate as test set. In this way, a separate set of recordings is used for model evaluation.

Site	**Total number of videos**	**Videos selected**
Vellore, India	355	67
Chicago, US	223	67
Norway	155	67

Table 4.1: The collection of infant videos

Although each video has a duration of 2 to 5 minutes, only 100 frames from each video are included in the dataset. The frames are randomly sampled aiming to cover most of the different infant poses present in a video. Overall, the CIMA Keypoints Dataset contains 20 100 frames. Table 4.2 gives an overview of the dataset with respect to the split between the different subsets.

	Fraction	**Videos**	**Frames**	**Keypoints**
CIMA Train	74%	149	14 900	104 300
CIMA Val	6%	12	1 200	8 400
CIMA Test	20%	40	4 000	28 000

Table 4.2: CIMA Keypoints Dataset

4.2.2 Annotation

To train a model on the extracted frames, information about the position of body parts had to be manually specified. This process, referred to as *annotation*, was performed as a collaboration with the research group at NTNU and St. Olavs Hospital. The annotation process was performed during a period of five weeks, with the annotation work being divided between five different annotators. When distributing data among the annotators, all samples of the test set were assigned to the independent annotators at St. Olavs Hospital.

Annotation Program

To facilitate a simple and efficient annotation process, we developed a program as an aid to perform and inspect annotations. The graphical interface was implemented using the Python programming language and the Tkinter GUI package [58], which made it possible to support both Mac and Windows operating systems. In the beginning of the annotation process, each annotator received a memory stick containing the program executable and the frames to be annotated.

In the annotation program, you are presented with one frame at a time. To start annotating a frame, you click on the location of the nose. This triggers a blue marker to appear at the location you believe the nose to be located. If there is need to adjust the location of the marker further, it can be dragged to a new location. When you are confident with the placement of the marker, the same procedure is followed for the remaining body parts. Each keypoint is assigned a marker in a predefined color, as illustrated in Figure 4.1. A screenshot visualizing the graphical interface of the annotation program is displayed in Figure 4.3.

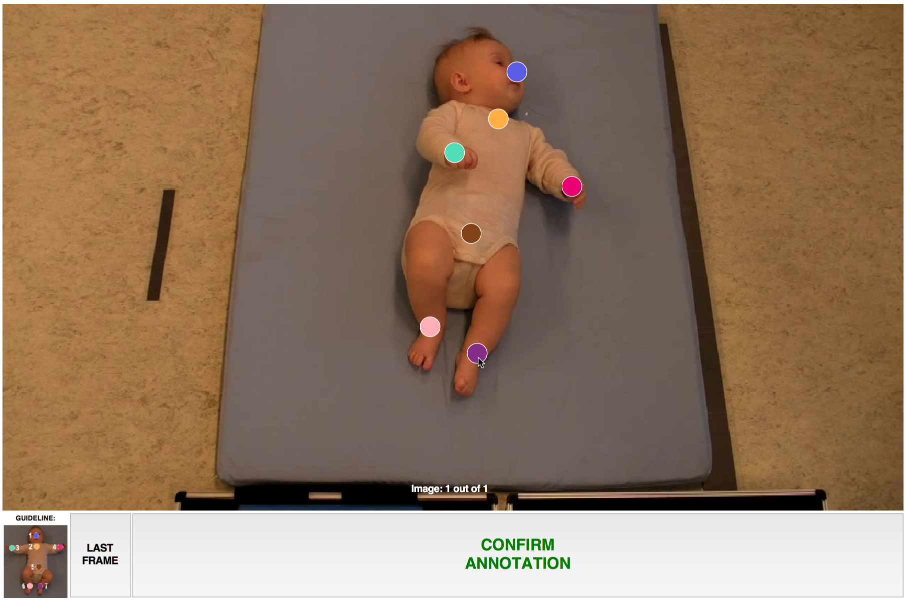


Figure 4.3: The annotation program utilized to specify body part locations

To ensure correct annotations, several helping steps were introduced. First, every time the annotators opened the program, they had to complete a training procedure containing five frames of different infant poses. In this phase, a wrongly placed point would yield a notice, requiring it to be corrected in order to proceed. Second, a guideline image is available to the annotator at all times in the lower left corner of the screen. By comparing the placement of the markers in an annotation frame with the guideline image, annotators could verify that keypoints were correctly placed.

Annotations from each session were saved in a tabular text format (CSV). This enabled simple data loading and processing, in addition to visual inspection using a spreadsheet program. To ensure no data was lost, keypoint locations were saved every time the annotation of a frame was confirmed. Upon program restart, progress thus continued where the annotator left off in the previous session.

Inter-Rater Reliability

To get a notion of the quality of the collected data, a set of frames is annotated by all annotators. In total 100 frames from four different videos are used for this purpose. By analyzing the variability between annotators, we gain knowledge about the consistency and quality of the obtained annotations.

For each keypoint in each frame, we use the mean of the individual annotations as the *ground truth*. Based on this, we calculate the error of each annotation using Euclidean distance. Finally, we evaluate relevant statistics such as mean and standard deviation of the errors, aggregated over annotators and body parts.

4.3 Baseline Models

Before developing our own model, we decided to research what level of performance can be achieved on our task using existing solutions. To work towards our goals of model precision and efficiency we decided to use CMU-Pose [1], which is recognized as a state-of-the-art model for real-time Human Pose Estimation.

As described in Section 3.2.2, this model uses the first parts of a VGG-19 model [43] as the feature extractor. Subsequently, the prediction is performed in two branches, as visualized in Figure 4.4. Branch 1 is responsible for keypoint localization, while the second branch is concerned with finding the connection between two keypoints. The latter is crucial for enabling multi-person Human Pose Estimation using this model. The two-branch block is repeated in six consecutive passes, referred to as stage 1 to t in the figure. Each block receives the output from the previous as its input. The original model is trained on the COCO Dataset, and is among the highest performing models on the COCO Keypoint Challenge [52]. In total, the CMU-pose architecture consists of 52.3 million parameters. For further details on this model, we refer to the original paper.

To evaluate our experiments, we define and implement three baseline models using the CMU-Pose architecture. The first, *Baseline I*, is loaded with weights released by the researchers behind the original CMU-Pose model. From the outputs of the model, we extract only the keypoints described in Figure 4.1. The results of *Baseline I* give insight into how a model trained solely on the general COCO Dataset can perform on the task of Infant Body Pose Estimation.

With *Baseline II*, the same model is used as a starting point. However, some of the weights of the network are fine-tuned for single-person Infant Body Pose Estimation using the annotated CIMA Keypoints Dataset. This reflects the second type of transfer learning, with a pre-trained model being used for initialization.

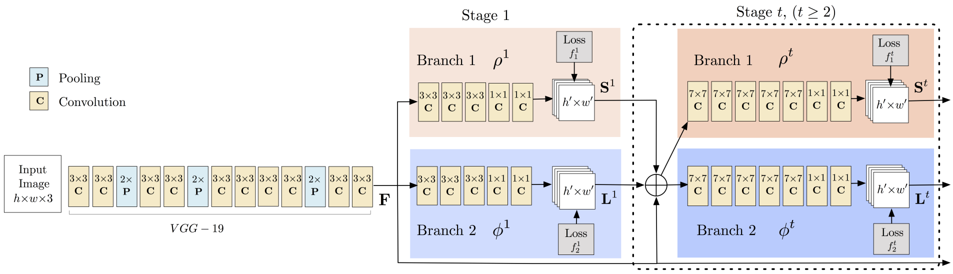


Figure 4.4: Illustration of the CMU-Pose architecture [1]

Finally, in *Baseline III* CMU-Pose is trained from scratch for single-person Infant Body Pose Estimation. The model is first trained on single-person images from the COCO Dataset, followed by fine-tuning using the CIMA Keypoints Dataset. The training procedure from this experiment is further used when proposing new network architectures.

4.4 Model Exploration

During this Master's project, not only has the state-of-the-art approaches related to our problem been assessed, we have put emphasis on researching what modifications can be made to construct a Deep Learning model solving the task of Infant Body Pose Estimation more seamlessly compared to existing approaches.

In this section, several suggestions for improvements are presented. A common theme of the proposed changes is the aim to improve run-time performance by reducing the number of parameters of the model. The modifications are to large extent inspired by the development in related areas of research, and make a basis for the experiments carried out during the project.

In Section 5.3, these modifications are evaluated with respect to the earlier discussed CMU-Pose model, both in terms of precision and efficiency.

4.4.1 Lighter Feature Extractor

The feature extractor is an essential part of the Deep Learning model by being responsible for extracting interesting properties from the images presented to the model. These properties are the basis for the subsequent stage of the network, namely the process of detecting body parts.

As illustrated by Table 3.1, following the progress of Deep Learning, more clever feature extractors have been proposed. Additionally, recent approaches have focused on extracting features using a less costly process, indicated by a lower number of parameters.

To obtain a system that is accurate and applicable in clinical practice, we aim to find a feature extractor that provides high-quality properties in a timely manner. We consider two existing feature extractors for this purpose.

MobileNet [46] is among the state-of-the-art feature extractors that focuses the most on run-time performance. However, as illustrated by the results on the ImageNet classification competition (Table 3.1), extensive focus on efficiency might come at the cost of decreased precision. Therefore, we explore how the lightweight feature extractor affect the quality of the subsequent process of detecting body parts.

As described in Section 3.2.1, DenseNet [2] is designed to deliver features promptly while preserving high performance. We therefore consider this approach relevant to achieve a proper trade-off between efficiency and precision.

In order to measure the marginal effects of different feature extractors on the task of localizing body parts, the network architecture of CMU-Pose is modified on the feature extraction part. Specifically, the partial VGG-19 network utilized in CMU-Pose is replaced with corresponding parts from DenseNet and MobileNet. Subsequently, the networks are tuned according to the same training scheme as *Baseline III*.

4.4.2 Single Branch

With CMU-Pose, the outputs of the model do not only express where the body parts are located, they also estimate the connections between body parts. Primarily, this provides a tool to distinguish between body parts of different people when there are multiple humans in an image.

Because the computation of part affinity fields takes significant processing time, we analyzed how much value this step provides in the single-person situation. Therefore, we experimented training the model without this part, having confidence maps as the only outputs.

4.4.3 Fewer Passes

An iterative detection process is valuable for several reasons. Besides mitigating the common Deep Learning problem of vanishing gradients through intermediate supervision, multiple passes might lead to more robust predictions being produced at the later iterations. The latter could be the case by learning from the knowledge extracted in previous passes. Hence, this could simulate some form of ensemble learning by obtaining the final conclusion based on several attempts.

However, the repeating process is a costly operation. We expected that the precision would increase with the number of passes but also that there was a point where detection performance would no longer improve. If this convergence occurs at a low number of passes, the cost of the network architecture could be reduced. Consequently, we would like to discover the number of passes where there is little to no improvement in the precision of the model by increasing the number of iterations further.

In the experiments, the CMU-Pose model was evaluated with the number of detection passes ranging from 1 to 6.

4.4.4 Dilated Kernels

With several detection passes, it is equally important to make each of these as efficient as possible in order to preserve low model latency. In ConvNets, large kernels enables output neurons to cover large portions of the input feature maps. This is useful when we want to pull out global context over a few convolutional layers. This is the case in the stage of body part detection.

While larger kernels reflect larger receptive fields, the run-time performance of convolutional layers highly corresponds with the size of the kernel utilized. With larger kernels the number of parameters increases significantly. Nevertheless, with dilated convolution the kernel size is reduced while still providing large receptive fields.

As a result, we analyzed the marginal effect of replacing convolutional layers utilizing large kernels with dilated convolutions. Specifically, the 7×7 convolutions of detection pass 2 through 6 of CMU-Pose was substituted with 4×4 dilated convolutions reflecting a dilation rate of 2. In this way, the size of the receptive field is preserved while reducing the number of parameters to less than half.

4.4.5 CMU-Pose Light

Based on the changes proposed in Sections 4.4.1 to 4.4.4, a lightweight CMU-Pose model is constructed. The model is referred to as CMU-Pose Light and aims to evaluate whether the various modifications can be combined to provide efficient and accurate body part detection.

First of all, CMU-Pose Light utilizes DenseNet [59] as the feature extractor. Secondly, the model incorporates the various proposed changes of the detection process. This includes reducing the number of detection passes to two, replacing expensive 7×7 convolutions with dilated convolutions and at the same time leaving out the prediction of part affinity fields.

Finally, in Section 5.3.5, CMU-Pose Light is compared with the original CMU-Pose model both in terms of precision and run-time performance.

4.5 CIMA-Pose Model

In Section 4.4, we described parts of the research that have been conducted for improving state-of-the-art approaches within Human Pose Estimation. The effort resulted in proposing a lightweight network architecture inspired by the CMU-Pose model as well as recent progress in the field of Deep Learning. Regardless of this, there has not been extensive focus on improving detection accuracy. In fact, most of the modifications that have been discussed so far have aimed to reduce the inference time of the model.

Nevertheless, in the upcoming section we introduce a self-proposed model with the goal to improve the quality of the predictions while at the same time accelerating run-time performance further. We refer to the model as CIMA-Pose.

4.5.1 Network Architecture

CIMA-Pose combines advances of Deep Learning to construct a remarkably neat and high-performing ConvNet. First of all, the model utilizes the recently proposed DenseNet [2] for extracting high-quality features in an efficient manner. Based on these set of features, an iterative detection process is performed, inspired by the success of state-of-the-art approaches within Human Pose Estimation [1,48,49,60]. Lastly, the obtained estimates of body part locations are upsampled to produce predictions of high precision. The overall architecture of CIMA-Pose is illustrated in Figure 4.5. The model consists of only 2 350 099 parameters.

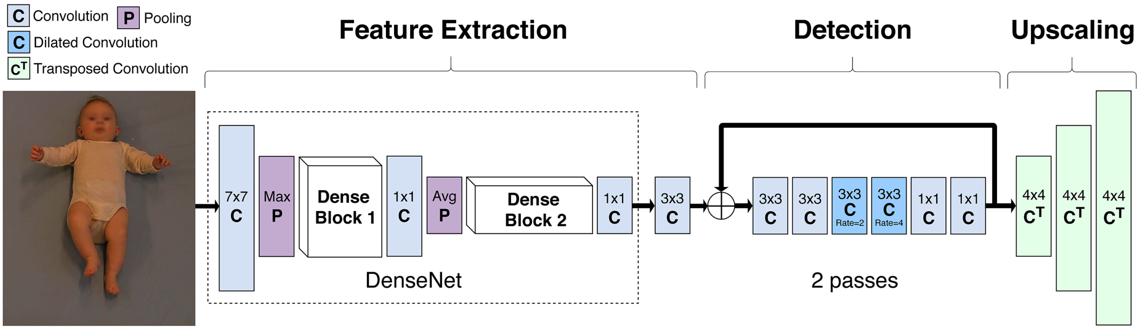


Figure 4.5: The network architecture of CIMA-Pose

4.5.2 Stages

The three parts of the model, referred to as *stages*, include *Feature Extraction*, *Detection* and *Upscaling*. In the upcoming section, each of these will be addressed in more detail.

Feature Extraction

Along the lines of transfer learning, a powerful feature extractor is desired for solving the complex task of Infant Body Pose Estimation using high-quality general knowledge of the content in images. The DenseNet architecture [2] is among the approaches that have proven particularly promising for this purpose. Additionally, as discussed in Section 3.2.1, DenseNet facilitates low memory consumption as well as high run-time performance.

As illustrated by Figure 4.6, the DenseNet architecture consists of a sequence of *dense blocks*. The dense blocks are made up of many convolutional layers and are characterized by the dense connectivity pattern improving the information flow in the network [2]. Each dense block is followed by transition layers containing 1 × 1 convolution and average pooling.

An interesting attribute with DenseNet is the ability for each of the dense blocks to produce robust features, but at different scales. Thus, it might not be necessary to utilize the whole sequence of dense blocks in order to extract representations of high quality. Building upon this observation, the feature extractor of CIMA-Pose is comprised of only the first two dense blocks of DenseNet. Consequently, an input image of size $H \times W$ is compressed into feature maps of size $\frac{H}{8} \times \frac{W}{8}$ as displayed by the height of the layers in Figure 4.5.

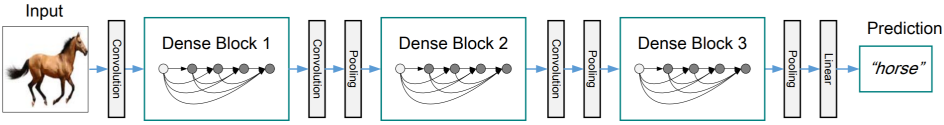


Figure 4.6: An abstract presentation of the DenseNet architecture [2]

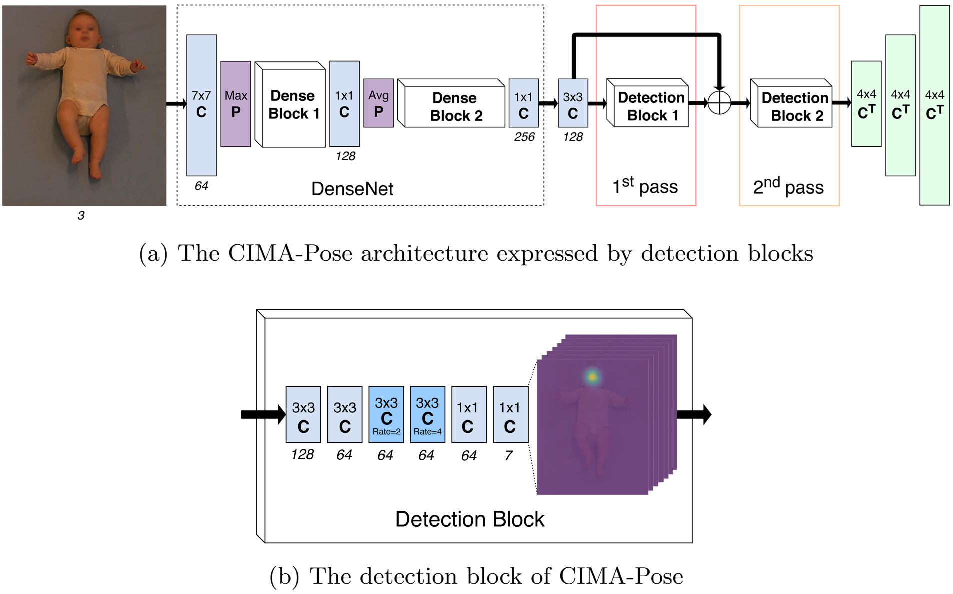
Besides consisting of the first building blocks of DenseNet, the feature extractor utilized in CIMA-Pose includes an additional convolutional layer. The purpose of this is two-fold. First, the features extracted by the DenseNet are of high resolution and may contain information excessive for the detection part of the network. Second, being pre-trained on ImageNet [22], DenseNet extract features with the aim to represent information specifically suited for the Image Classification task. As a result, we are interested in training a convolutional layer that is able to reduce the amount of information provided by DenseNet and at the same time deliver features to the detector that are adapted for the task of Human Pose Estimation.

Detection

The detection stage is concerned with estimating where the body parts reside in the provided image. In order to perform this task accurately, the process of predicting body part locations is repeated. However, given the powerful detector of CIMA-Pose we require estimates to be computed solely two times.

The network architecture of Figure 4.5 expresses this iterative process by displaying the computation of each detection pass being dependent on the output of the previous pass as well as the pool of features produced by the feature extractor.

Each *detection block* (Figure 4.7b) contains the same configuration of convolutional layers, but has its own unique set of weights. This is made explicit in Figure 4.7a, where the detection blocks are included. Utilizing a combination of dilated convolution and standard convolution, the detector manages to sufficiently aggregate information from the input feature maps and previous detection pass using a low number of parameters.



(a) The CIMA-Pose architecture expressed by detection blocks

(b) The detection block of CIMA-Pose

Figure 4.7: Explicitly displaying the detection blocks of CIMA-Pose

The detector is initiated with two standard 3 × 3 convolutions. The first of these layers is mainly interested in detecting which parts of the input is relevant for the task we are dealing with. On the other hand, the second convolutional layer aims to compress the information even further, expressed by the reduction in the number of feature maps from 128 to 64.

Subsequently, dilated convolutions are carried out. This includes two layers of 3 × 3 kernels. The layers differ in the amount of dilation that is performed, reflected by a dilation rate of 2 and 4 respectively. The dilated convolutions can be considered the core of the detector by gathering global information relevant for predicting body part locations. This is obtained by the large receptive fields resulting from dilated convolution.

In the final phase of the detection pass, the outcome of the second dilated convolutional layer is processed by two standard 1 × 1 convolutional layers. These takes the final decision of predicting where the body parts are located within the image. The first 1 × 1 convolution is applied to represent the knowledge obtained through the dilated convolutions on a format that is suitable for distinguishing between different body parts in the final convolutional layer. The subsequent layer itself is concerned with the actual production of confidence maps. The detector outputs seven confidence maps, one per body part. With CIMA-Pose, we do not consider additional output necessary, as opposed to models such as CMU-Pose [1].

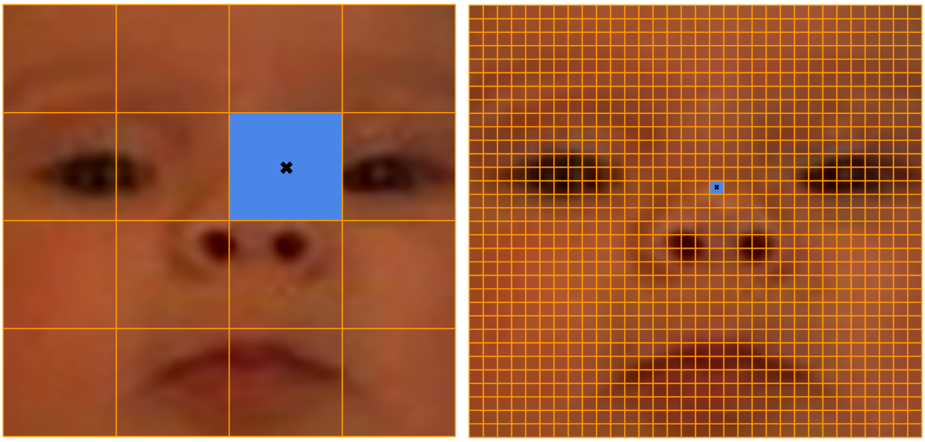
Upscaling

The outputs of the detection blocks are computed based on the resolution obtained by the feature extractor. Correspondingly, the confidence maps are of low spatial dimensionality which restricts the capability of the detector to generate estimates of high precision. In other words, the confidence maps produced by the detector are of low resolution compared to the input image and thus the network can not make predictions that are accurate on the pixel-level with respect to the input image space.

Accordingly, an upscaling stage is proposed, mapping the low-resolution output $(\frac{H}{8} \times \frac{W}{8})$ of the confidence maps onto a space with the same resolution $(H \times W)$ as the input image. In more details, the predictions of the final detection pass are upsampled using a series of three transposed convolutions. Each of these layers is responsible of increasing the size of the confidence maps by a factor of 2. This produces predictions in a higher-resolution space by utilizing the knowledge of the original outputs.

In similar fashion to Long et al. [38], the weights of the upsampling layers are fixed to values resembling bilinear interpolation. This is performed as an efficient way to utilize information from several nearby values in the low-resolution space to produce estimates of higher precision. The transposed convolutions utilize kernels of size 4×4 to obtain estimates that are based on the values of surrounding neurons. Thus, the value of a single neuron in the resulting confidence maps will be computed from the values of four neighboring neurons in the low-resolution confidence maps.

As illustrated by Figure 4.8, the upscaling procedure enables the model to perform more accurate predictions, reflected by the size of the regions the model is able to distinguish between. The figure displays parts of a 360 × 640 sized image captured according to the standard video set-up discussed in Section 4.2.1.



(a) The precision provided by the confidence maps of the detector

(b) Upscaling improves the level of precision for predictions

Figure 4.8: Visualizing the effect of upscaling

4.5.3 Implementation Details

The network architecture of CIMA-Pose has been developed with the Python Deep Learning library of Keras [61]. Moreover, the TensorFlow [62] backend has been utilized for implementing functionality and concepts not provided directly through Keras.

4.6 Training Strategy

To train a neural network, several aspects must taken into consideration. First, the data the model takes as input and aim to produce as output should be clearly defined. Second, the way optimization is performed should be determined. In this section, the training strategy being followed while carrying out experiments is discussed. This also includes more extensive explanations of how the weights of the CIMA-Pose model has been obtained.

4.6.1 Data Preparation

In the experiments, we ensure that the images are presented to the model with the same height and width. As a result of images varying in aspect ratio as well as the original resolution, resizing and padding operations are performed before the images are inputted to the model. This ensures that input images have a height and width of 360 and 640 respectively. Although the input resolution is fixed during training, following the nature of ConvNets predictions can be made for images of any size.

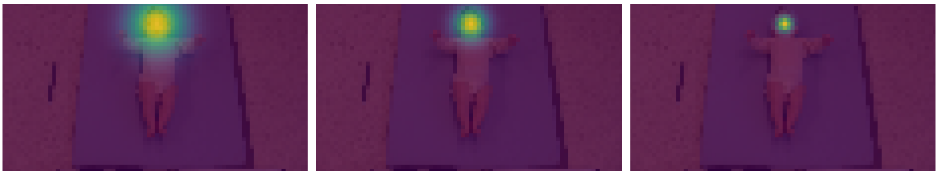
Before being presented to the model, random transformations are performed on each of the images. This process, called data augmentation, includes rotating, scaling, and shifting the images vertically and horizontally. Introducing these random distortions, the model has the capability of better learning the general patterns that are important to solve the task. In addition to performing data augmentation, the order images are presented to the model is randomized. This makes the model more robust by every time letting the weights be updated based on performance on unique batches of images.

Having constructed input images, it is important that the expected outputs are of appropriate format. First, desired confidence maps are produced based on the same random transformations as the input images. This should ensure that there exists a correct mapping from images being presented to the model and the confidence maps defining where body parts reside in the respective images.

When it comes to the generation of confidence maps, the model should be rewarded according to how close a prediction is to the actual body part location p^*. The values of the seven matrices defining the confidence maps are set to floating point numbers between 0 and 1. The value of a pixel p in a target confidence map C^* is defined according to Equation 4.1. A high value reflects close proximity to the actual location.

$$C^*(p) = exp(-\frac{\|p - p^*\|}{\sigma^2}) \tag{4.1}$$

While producing confidence maps, the size of the regions with values higher than 0 should be determined. The size of the region is defined by the parameter referred to as σ. Figure 4.9 displays confidence maps produced from different values of σ. In the figure, purple reflects a confidence value close or equal to 0, while yellow represents the optimal keypoint location.



(a) $\sigma = 7$ (b) $\sigma = 4$ (c) $\sigma = 2$

Figure 4.9: Target confidence maps representing the location of the nose.

4.6.2 Optimization

The optimization process is performed using Stochastic Gradient Descent in combination with momentum. A momentum value of 0.9 is used.

As loss function, Euclidean loss is used, as defined in Equation 4.2. Calculated over a batch of samples, N is the number of confidence maps, $C_i \in R^{45\times 80}$ is a ground truth confidence map, and $C_i^* \in R^{45\times 80}$ is the corresponding predicted confidence map. The loss is calculated after each detection pass, only two times with the CIMA-Pose model.

$$L_{\text{euclidean}} = \frac{1}{2N}\sum_{i=1}^{N} \|C_i^* - C_i\|_2^2 \tag{4.2}$$

As another trick to facilitate a quick and stable learning process, we first train the network using σ of 7, and step-wise decrease it towards a value of 2. Intuitively, this lets the model learn to perform coarse predictions, before focusing on smaller details.

An additional consideration is how much the weights are updated in the different phases of the training process. After initializing the model, a low learning rate is used to avoid exploding gradients. Subsequently, in coherence with decreasing values of σ, the learning rate is increased. In the final phase of the training process, a lower learning rate is used to fine-tune the weights.

When it comes to the data the models are trained on, the process is carried out in two steps. As described in Section 4.3, weights are first optimized on the general COCO Dataset before learning patterns specific to the task of Infant Body Pose Estimation using the self-developed CIMA Keypoints Dataset. This strategy enables us to utilize the large COCO Dataset to achieve a general and stable pose estimator, and to avoid overfitting in the subsequent training phase. In initial experiments, we observed that the models with only one detection branch or fewer passes learned quicker in the first training step. For these models, we therefore train for a fewer number of epochs on the COCO Dataset.

4.6.3 Training of CIMA-Pose

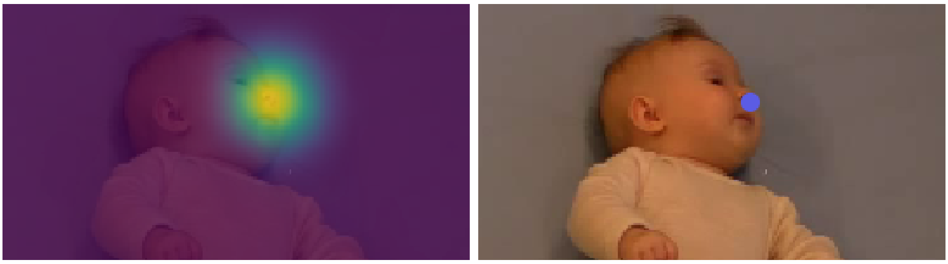
In general, the weights of the CIMA-Pose model are adjusted according to the same procedure as discussed in Section 4.6.2. However, with respect to the individual parts of the model, there are some differences.

First of all, the weights of the pre-trained layers of the feature extraction stage are not modified further. On the other hand, the final convolutional layer of the feature extractor is trained on the COCO Dataset in order to deliver features suited for Human Pose Estimation. These weights are fixed while training the model on the CIMA Keypoints Dataset. The detection blocks of CIMA-Pose are optimized using feedback both from the COCO and CIMA datasets. Finally, as mentioned in Section 4.5.2, the transposed convolutional layers responsible of performing upscaling do not require any training at all as the values of the kernels are fixed to bilinear interpolation.

4.7 Extraction of Coordinates

After performing the upscaling, the model outputs seven confidence maps of size 360 × 640, one for each body part. These matrices are interpreted as a spatial representation of the model's *confidence* in the location of a body part.

To extract coordinates of a keypoint, the maximum value of each confidence map is located. Dividing the x and y indices of this value on the width and height of the confidence map respectively, the relative coordinates of the keypoint are obtained. In Figure 4.10b, we illustrate the extracted keypoint location from the confidence map of Figure 4.10a.



(a) Model output

(b) Extracted keypoint

Figure 4.10: Extracting keypoint location from a confidence map

Finally, the coordinates of the keypoints in the different frames of a video are collected and stored in tabular format in a CSV file. The coordinates are presented in a way similar to Table 4.3.

Nose$_x$	**Nose**$_y$
0.5383	0.1433

Table 4.3: The coordinates of the extracted keypoint of Figure 4.10b

4.8 Evaluation Metrics

In order to evaluate the precision of the proposed models, we define a set of metrics. In contrast to the loss function, these metrics are evaluated using the predicted keypoint coordinates. We define two metrics, which measure two different qualities of the models.

4.8.1 Mean Keypoint Error

Mean Keypoint Error (MKE) is a single, continuous number which describes the average Euclidean distance in pixels between the predicted keypoints and the ground truth values of a 360×640 sized image. The lower this value is, the better a model performs. By calculating the same metric on the inter-rater samples, we can compare a model to human performance.

As described in Section 4.2.1, the height of the mattress in the CIMA recordings (90 centimeters) can be used as an estimate for measuring the actual distance within a video frame of the CIMA Keypoints Dataset. Based on this presumption, in images of height 360 a distance of 4 pixels corresponds with 1 cm ($\frac{360}{90cm} = \frac{4}{1cm}$). This estimate can be used to get a notion of the MKE measure in terms of centimeters.

4.8.2 Keypoint Accuracy

In Human Pose Estimation, a common practice is to measure accuracy using the Percentage of Correct Keypoints (PCK) [63]. The scale of the human can be taken into consideration when defining a correct keypoint, for instance by defining a threshold relative to the body. As an example, the MPII competition [55] uses a version called PCK_h, where the threshold is set to 50% of the head segment.

We use a slightly modified version of the PCK_h metric, having the distance from the nose to the upper chest as the basis for the evaluation. We refer to this as PCK_i. Instead of using a fixed 50% fraction of this segment, we evaluate the metric for several thresholds corresponding with fractions of 50%, 30% and 15%. The different measures provide evaluation of the model accuracy in different levels of detail.

The threshold values are illustrated in Figure 4.11, where the black line outlines the nose-chest segment. In this example, with a threshold of 50% (PCK_i@0.5) any nose prediction within the yellow circle is considered correct. With the smallest 15% threshold, only predictions within the purple region are accepted.

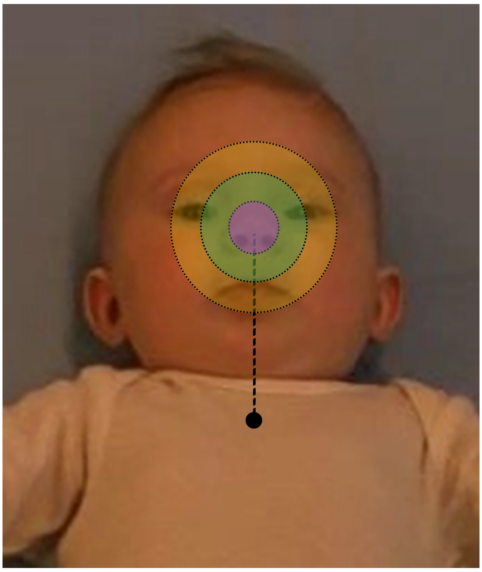


Figure 4.11: Different thresholds for measuring keypoint accuracy (PCK_i@0.5 - yellow, PCK_i@0.3 - green, PCK_i@0.15 - purple)

4.9 Tracking

The CIMA-Pose model makes predictions from a single image. Accordingly, when applied to a video in a frame-by-frame manner, the relationship between consecutive frames is not taken into account.

To take this aspect into consideration, we experiment using a simple smoothing algorithm called Moving Average. As defined in Equation 4.3, an estimate is represented as the average of the current predicted value, and its nearby predictions in time. In the equation, n refers to the size of the window that is utilized while p_{t+i} denotes the raw estimate of the corresponding keypoint at time $t+i$. Applying this filter to a time series, small fluctuations in the raw predictions will be effectively smoothed out as displayed by Figure 4.12. If one data point has a large difference from surrounding values, this will result in a spike in the resulting signal. In this project, a window size of 5 is used.

$$\bar{p_t} = \frac{1}{n} \sum_{i=-\lfloor n/2 \rfloor}^{\lceil n/2 \rceil - 1} p_{t+i} \tag{4.3}$$

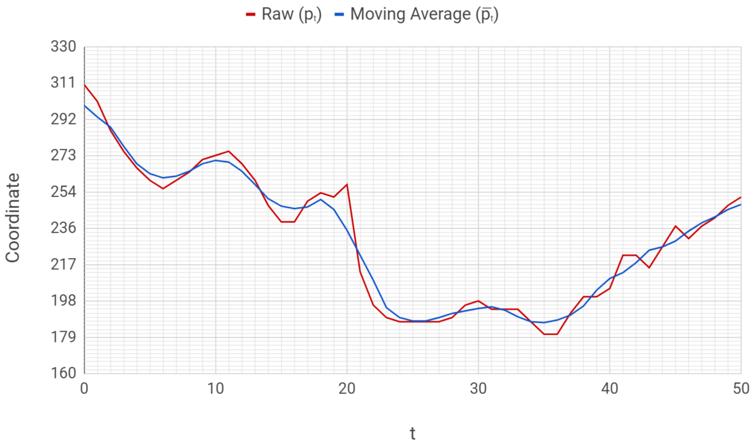


Figure 4.12: The effect of applying Moving Average to a raw signal

Although there are many alternatives to the simple Moving Average algorithm, smoothing has not been the main focus of this project. Therefore, more intricate methods are not evaluated.

Based on the results obtained with applying smoothing, annotated videos displaying tracking of body parts can be easily generated. This might be valuable both for inspecting the performance of the method as well as an aid for clinicians to properly observe the movements that are relevant for detecting CP.

As a qualitative evaluation, GMA expert Dr. Lars Adde ranked the method's ability to track body parts on 30 second extracts from each of the 40 recordings of the CIMA Test set. On a scale from 1 to 5, the videos were scored according to the following assertion: *"The points follow the body parts in a good way"*. His evaluations are described in Section 5.4.7.

Chapter 5

Results

Having described the proposed model and research process in Chapter 4, the upcoming chapter describes the results that were obtained.

In our experiments, the self-developed CIMA Keypoints Dataset has played a crucial role, both in training of new models, and evaluation of existing ones. Section 5.1 focuses on the quality of the annotated data, using the inter-rater samples from the CIMA recordings. Subsequently, in Sections 5.2 to 5.4, we evaluate the performance of the baseline models, the proposed hypotheses, and the final CIMA-Pose model. Models are evaluated using the CIMA Test set, which was kept aside until all experiments were completed. More specifically, models are scored in terms of error (MKE), accuracy (PCK_i) and run-time performance with respect to frames per second (FPS). Progress during training is displayed through learning curves, comparing the loss on CIMA Val across different models.

In this project, all experiments and evaluations are carried out on a workstation, using a 2.20GHz CPU (Intel Xeon E5 v4) and a NVIDIA Tesla P100 GPU with 16GB of VRAM.

5.1 Annotation Consistency

As described in Section 4.2.2, a total of 100 frames originating from 4 videos were annotated by all annotators, to be used as basis for evaluating the inter-rater reliability of the CIMA Keypoints Dataset. The box plot in Figure 5.1 visualizes how the keypoint error varied with respect to each annotator, with mean values ranging from 1.84 pixels with annotator 3 to 2.54 pixels of annotator 5.

Figure 5.2 shows a box plot with respect to each body part, illustrating that the annotation differences were significantly higher and more varied for the keypoint concerned with the hip center than the nose. However, we see that the mean and variance across the other body parts are consistent. Table 5.1 summarizes the Mean Keypoint Error of each annotator across different body parts. The average MKE values of 3.34 for the hip center, and 1.17 for the nose, concretize the visual impression given by Figure 5.2. Finally, Figure 5.3 shows the differences between annotators by providing an example of the five annotations of the keypoint belonging to the right wrist.

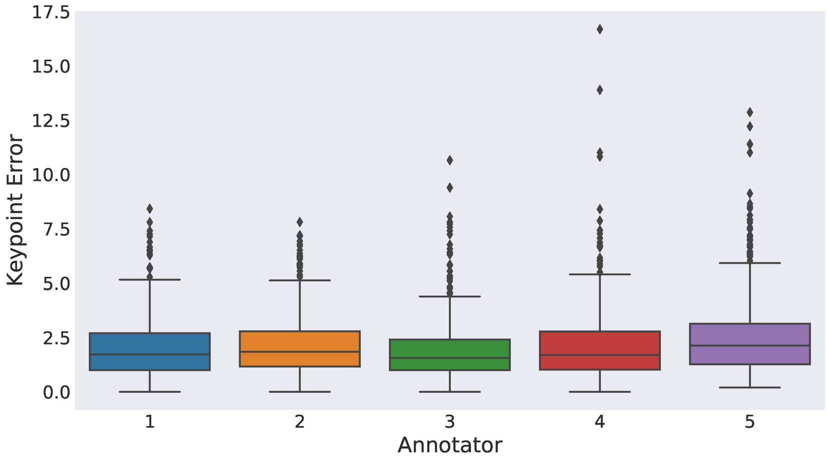


Figure 5.1: Variability across annotators

Table 5.1: MKE across body parts for each annotator

Annotator	MKE_N	MKE_{UC}	MKE_{RW}	MKE_{LW}	MKE_{HC}	MKE_{RA}	MKE_{LA}
1	0.99	**1.97**	2.57	2.68	**2.46**	**1.70**	1.83
2	1.53	2.36	**1.58**	1.82	2.76	2.41	2.41
3	0.93	2.04	1.68	**1.65**	2.99	1.83	**1.77**
4	**0.92**	2.47	2.00	1.73	4.05	1.71	1.78
5	1.48	2.06	2.06	2.23	4.44	2.88	2.66

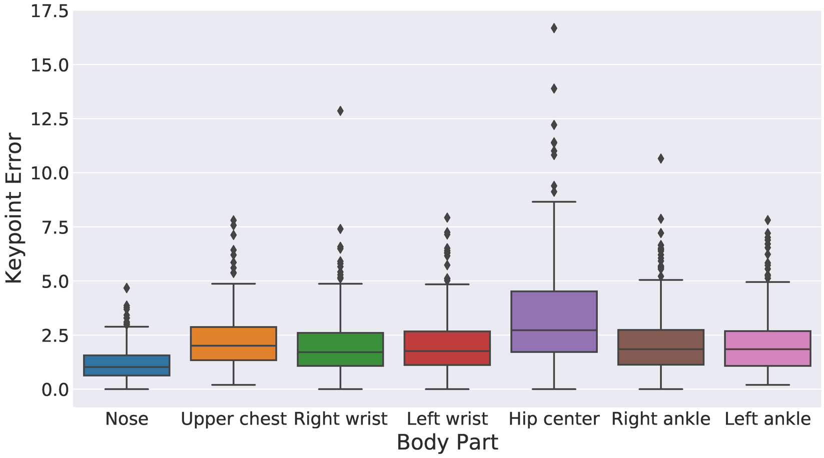


Figure 5.2: Annotation variability across body parts

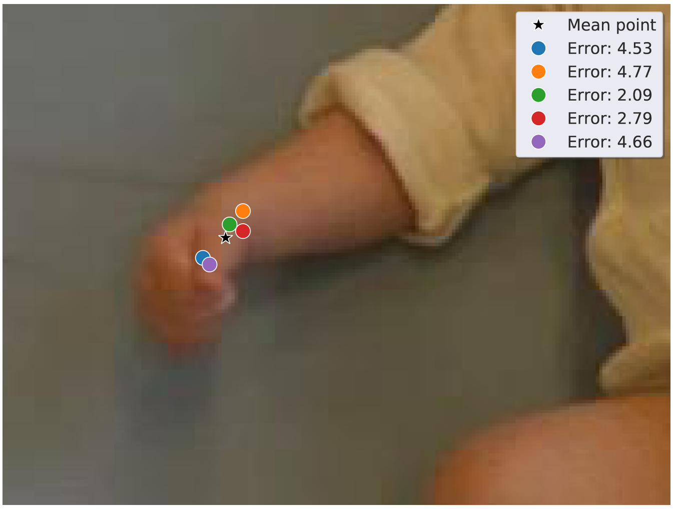
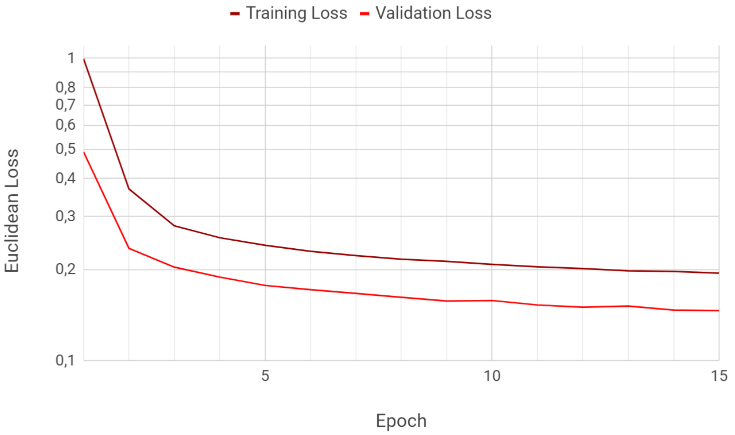


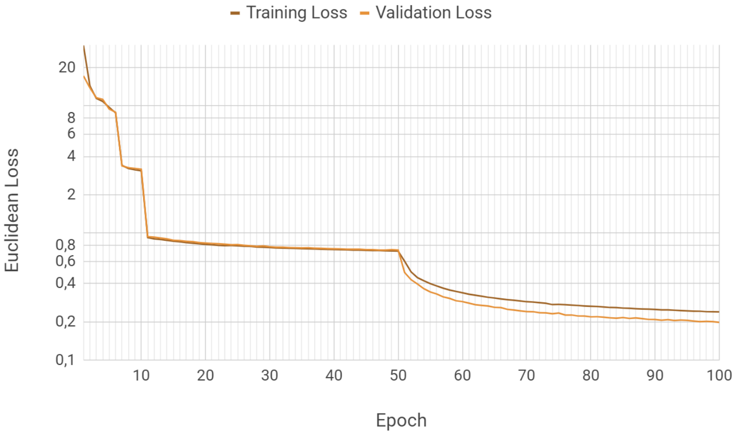
Figure 5.3: Differences across annotators

5.2 Baseline Performance

The baseline models are trained according to the strategies described in Section 4.3. Figure 5.4 displays the improvement of the models over the course of training with respect to Euclidean Loss. As seen in the charts, at the end of the training process the loss has stopped decreasing for both Baseline II and Baseline III. Also interesting to notice, the validation loss is lower than the training loss.



(a) Learning curve of Baseline II



(b) Learning curve of Baseline III

Figure 5.4: The progress of Baseline II and Baseline III measured by the Euclidean Loss of the confidence maps of the final detection pass

Table 5.2 summarizes the performance of the baseline models, both with respect to precision and efficiency. We observe that both Baseline II and Baseline III obtain better results on both, as well as in terms of run-time performance.

Table 5.2: Performance of baseline models

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
Baseline I	52 311 446	7.12	97.1	93.6	60.9	18.8
Baseline II	50 139 594	**4.48**	**99.7**	**98.3**	**73.5**	**20.0**
Baseline III	50 139 594	4.98	99.4	96.7	67.1	**20.0**

5.3 Evaluation of Model Exploration

In the upcoming section, the ideas described in Section 4.4 are evaluated. The proposed modifications are implemented as changes to the CMU-Pose model and evaluated after training the obtained ConvNet in a similar fashion as Baseline III.

5.3.1 Lighter Feature Extractor

Using MobileNet and DenseNet as feature extractors, Figure 5.5 displays the validation loss during training. During the first 50 epochs, the models are trained on the COCO Dataset, followed by additional 50 epochs of training on the CIMA Keypoints Dataset. With respect to MobileNet, we observe that the loss stops decreasing when the model is optimized on CIMA data. While both models achieve better run-time efficiency than Baseline III (Table 5.3), the improvement is more significant with MobileNet as feature extractor. However, this model also has higher MKE and lower accuracy values. The DenseNet-based model is closer to Baseline III in terms of MKE and accuracy.

Table 5.3: Performance of models utilizing different feature extractors

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
Baseline III (VGG-19 [43])	50 139 594	**4.98**	**99.4**	**96.7**	**67.1**	20.0
DenseNet [59]	44 539 082	5.47	99.2	95.2	60.9	24.8
MobileNet [46]	43 231 306	23.78	86.6	75.7	38.1	**27.0**

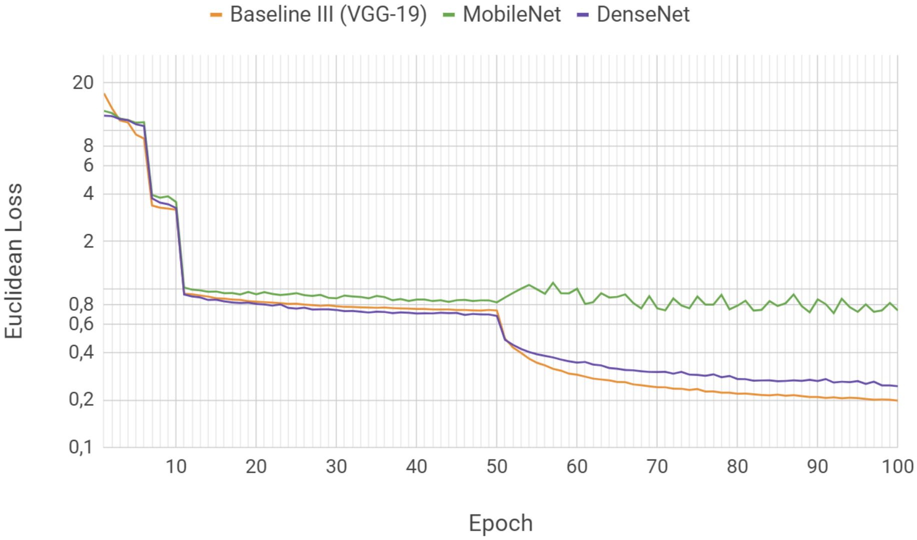


Figure 5.5: Learning curves of models utilizing different feature extractors

5.3.2 Single Branch

In Table 5.4, the model trained without the branch responsible for finding connections between body parts is compared with Baseline III. It can be observed that all precision metrics were improved by using only a single prediction branch. In addition, a 44% decrease of parameters and a 48% improvement in run-time are achieved.

Table 5.4: Performance of model only outputting confidence maps

Model	# Parameters	*MKE*	$PCK_i@0.5$	$PCK_i@0.3$	$PCK_i@0.15$	FPS
Baseline III	50 139 594	4.98	99.4	96.7	67.1	20.0
Single Branch	28 233 066	**4.58**	**99.7**	**97.9**	**72.1**	**29.6**

5.3.3 Fewer Passes

Figure 5.6 and Table 5.5 illustrate how the gain of Mean Keypoint Error flattens out after the first two passes of detection.

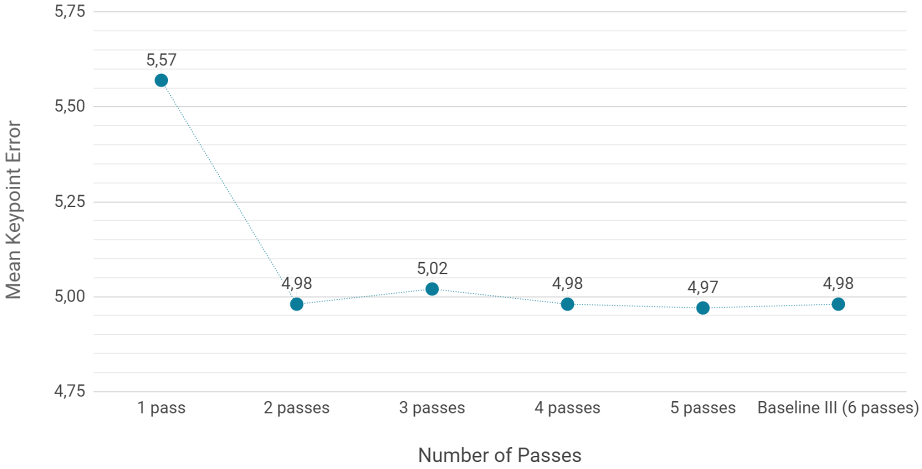


Figure 5.6: Correspondence between MKE and number of detection passes

Table 5.5: Performance of models with varying number of detection passes

Model	# Parameters	*MKE*	PCK_i@0.5	PCK_i@0.3	PCK_i@0.15	FPS
Baseline III (6 passes)	50 139 594	4.98	**99.4**	**96.7**	67.1	20.0
5 passes	41 785 651	**4.97**	**99.4**	96.6	**67.3**	22.5
4 passes	33 431 708	4.98	**99.4**	96.6	67.2	26.5
3 passes	25 077 765	5.02	**99.4**	96.6	66.6	32.1
2 passes	16 723 822	4.98	**99.4**	96.5	**67.3**	40.7
1 pass	8 369 879	5.57	98.8	94.7	60.9	**55.6**

5.3.4 Dilated Kernels

The effect of replacing the standard 7 × 7 convolutions of detection pass 2 to 6 of CMU-Pose with 2-dilated 4×4 convolutions is displayed in Table 5.6. With MKE of 5.30, the obtained model does not completely reach the level of Baseline III. However, in terms of efficiency, a relative improvement of 8.2 FPS is achieved.

Table 5.6: Performance of model utilizing dilated kernels

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
Baseline III	50 139 594	**4.98**	**99.4**	**96.7**	**67.1**	20.0
Dilated Kernels	22 134 474	5.30	**99.4**	95.7	62.3	**28.2**

5.3.5 CMU-Pose Light

The experiments carried out in the preceding sections (Sections 5.3.1 to 5.3.4) are combined to comprise CMU-Pose Light. Table 5.7 displays the performance achieved by this model after training. Noteworthy, the run-time efficiency is improved by 600%. However, the lighter model shows a higher MKE, and lower accuracy at the 30% and 15% thresholds.

Table 5.7: Performance of CMU-Pose Light compared to Baseline III

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
Baseline III	50 139 594	**4.98**	99.4	**96.7**	**67.1**	20.0
CMU-Pose Light	3 595 470	5.43	**99.6**	96.0	59.5	**112.4**

5.4 CIMA-Pose

5.4.1 Optimization

The weights of the CIMA-Pose model are tuned according to the training strategy discussed in Section 4.6. In particular, the network is trained for 15 epochs on the COCO Dataset, followed by 50 epochs on the CIMA Keypoints Dataset. The learning curve of CIMA-Pose is provided in Figure 5.7. Figure 5.8 displays how the model optimizes in comparison to the other experiments carried out. The discontinuity of the graphs of the lightweight architectures (CMU-Pose Light, Single Branch, and CIMA-Pose) reflects the decreased number of epochs these models are trained on the COCO Dataset. The figure shows that CIMA-Pose reaches a low validation loss compared to the other models.

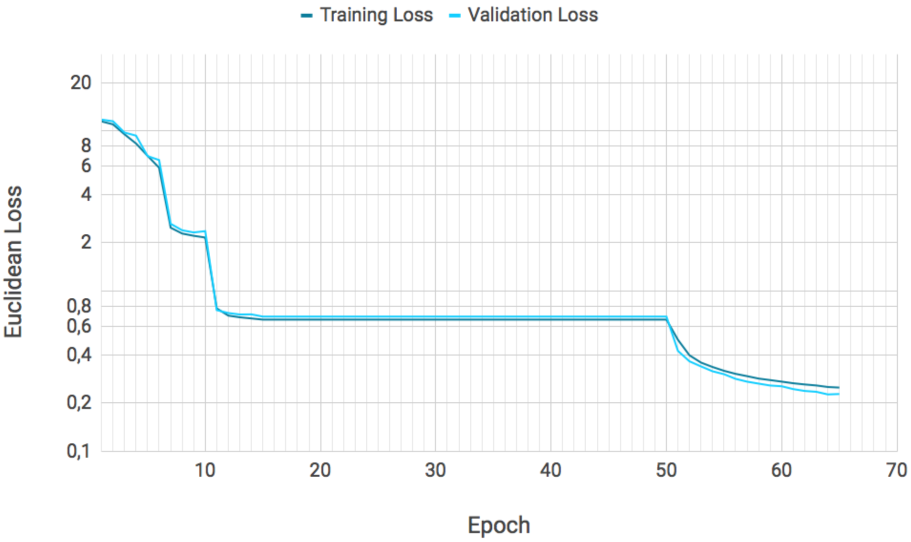


Figure 5.7: Learning curve of the CIMA-Pose model

5.4.2 Model Precision

Based on the obtained weights of CIMA-Pose, the precision of the model is evaluated with respect to the different measures described in Section 4.8. Table 5.8 displays how the model performs compared to the baseline models and CMU-Pose Light, reaching MKE of 3.81. As visualized in Figure 5.9, CIMA-Pose delivers top performance in all three accuracy measures. Most remarkably, CIMA-Pose achieves an accuracy of 82.2 on $PCK_i@0.15$ compared to 73.5 for the best performing baseline model.

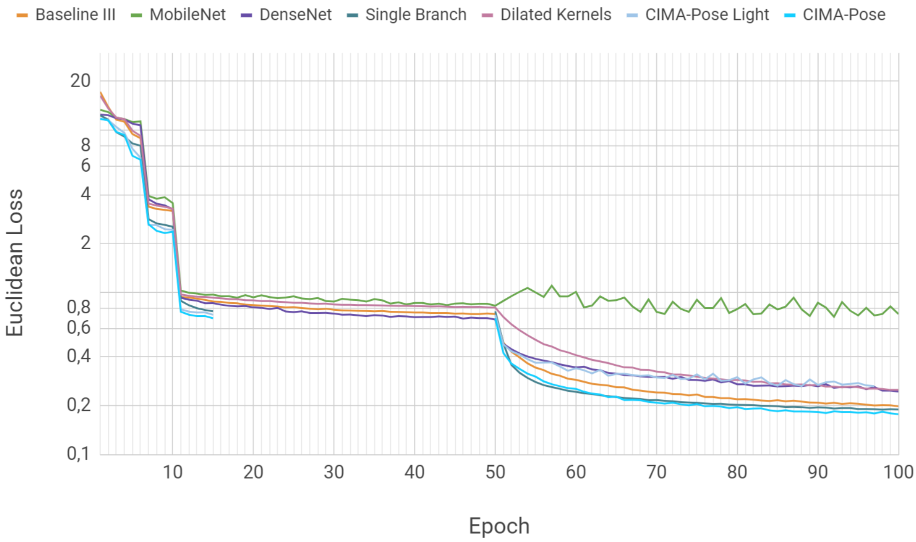


Figure 5.8: Learning curves of the experiments carried out

Table 5.9 presents how the models differ in performance with respect to individual body parts, and shows that CIMA-Pose outperforms both the baseline models and CMU-Pose Light. Comparing the values, we see that the keypoint of the nose has the lowest MKE, at 2.62. Among the other body parts, the errors are higher, with MKE varying from 3.69 for the left ankle to 4.50 with the keypoint of the upper chest.

Table 5.8: Precision of CIMA-Pose

Model	MKE	$PCK_i@0.5$	$PCK_i@0.3$	$PCK_i@0.15$
Baseline I	7.12	97.1	93.6	60.9
Baseline II	4.48	**99.7**	**98.3**	73.5
Baseline III	4.98	99.4	96.7	67.1
CMU-Pose Light	5.43	99.6	96.0	59.5
CIMA-Pose	**3.81**	**99.7**	**98.3**	**82.2**

To assess how CIMA-Pose performs on the validation and test data compared to the images the model has been trained on, evaluations are carried out and provided in Table 5.10.

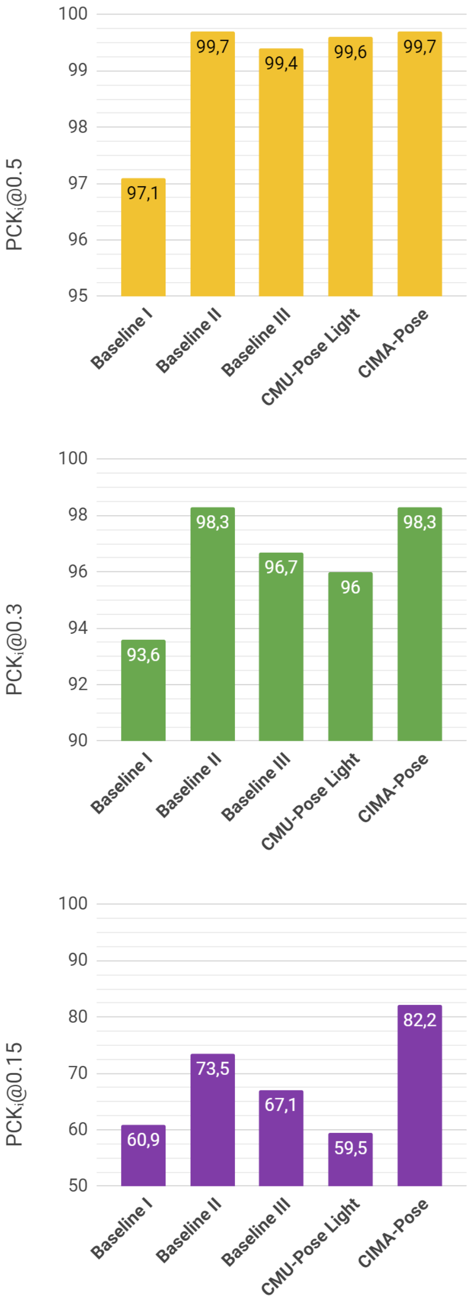


Figure 5.9: Comparison of precision at different thresholds

Table 5.9: MKE evaluated with respect to different body parts

Model	MKE_N	MKE_{UC}	MKE_{RW}	MKE_{LW}	MKE_{HC}	MKE_{RA}	MKE_{LA}
Baseline I	4.38	4.77	10.23	9.94	7.14	6.74	6.67
Baseline II	3.68	4.59	4.87	4.26	5.02	4.50	4.44
Baseline III	4.04	5.32	5.38	4.99	5.19	4.99	4.98
CMU-Pose Light	4.26	5.83	6.10	5.92	6.07	4.98	4.84
CIMA-Pose	**2.62**	**4.50**	**4.32**	**3.73**	**4.03**	**3.80**	**3.69**

Table 5.10: Precision of CIMA-Pose on the different portions of the CIMA Keypoints Dataset

Dataset	Size	MKE	PCK_i@0.5	PCK_i@0.3	PCK_i@0.15
CIMA Train	14 900	**3.40**	**99.7**	**98.8**	**87.4**
CIMA Val	1 200	3.96	99.4	97.4	81.4
CIMA Test	4 000	3.81	**99.7**	98.3	82.2

5.4.3 Run-Time Performance

When it comes to the efficiency of the CIMA-Pose model, the inference time of the network is evaluated. Table 5.11 presents the results of the run-time experiments that are conducted and displays the relative improvement of the different models with regard to Baseline I. It can be seen that the architecture of CIMA-Pose is both lighter and quicker than the models it has been evaluated against. Compared to the original CMU-Pose model (Baseline I), CIMA-Pose improves run-time performance by 660%. More specifically, a performance boost from 18.2 FPS to 120.8 FPS is achieved. Noteworthy, this is even higher than what was obtained with the earlier discussed CMU-Pose Light model, which operates at 112.5 FPS.

Table 5.11: Run-Time Performance of CIMA-Pose

Model	# Parameters	FPS	Improvement
Baseline I	52 311 446	18.2	1.0×
Baseline II	50 139 594	20.0	1.1×
Baseline III	50 139 594	20.0	1.1×
CMU-Pose Light	3 595 470	112.5	6.2×
CIMA-Pose	2 350 099	**120.8**	**6.6**×

5.4.4 Variations of CIMA-Pose

In order to get a notion of whether the network architecture of CIMA-Pose is chosen appropriately, minor changes to the CIMA-Pose model is implemented and subsequently evaluated. These variations include decreasing or increasing the number of detection passes as well as varying the degree of upscaling. Table 5.12 confirms that two passes achieve the highest precision. From Table 5.13 we observe that upscaling results in improved precision both in terms of MKE and the three accuracy metrics. Nevertheless, it is also shown that CIMA-Pose without upscaling reaches MKE of 4.90, corresponding to a slight improvement over Baseline III.

Table 5.12: Performance corresponding with varying number of detection passes

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
3 passes	2 655 637	3.86	**99.7**	98.2	81.7	115.2
CIMA-Pose (2 passes)	2 350 099	**3.81**	**99.7**	**98.3**	**82.2**	120.8
1 pass	2 042 188	4.09	99.6	97.6	78.6	**125.0**

Table 5.13: Performance of CIMA-Pose compared to models with fewer upsampling layers

Model	# Parameters	*MKE*	PCK_i*@0.5*	PCK_i*@0.3*	PCK_i*@0.15*	FPS
No upscaling	2 347 726	4.90	99.6	97.2	67.5	**138.9**
1 layer	2 348 517	4.09	**99.7**	98.1	79.1	133.3
2 layers	2 349 308	3.87	**99.7**	98.2	81.7	129.9
CIMA-Pose (3 layers)	2 350 099	**3.81**	**99.7**	**98.3**	**82.2**	120.8

Table 5.14: Performance of humans and CIMA-Pose on inter-rater samples

	Keypoint Error		Relative Keypoint Error	
	Human	CIMA-Pose	Human	CIMA-Pose
Mean	2.13	3.06	6.23%	8.80%
Std.	1.53	2.85	4.35%	7.29%
Min	0.00	0.00	0.00%	0.00%
50% quantile	1.80	2.42	5.24%	7.18%
75% quantile	2.78	3.62	8.21%	10.53%
95% quantile	4.84	7.45	14.60%	20.06%
Max	16.68	31.35	36.05%	69.37%

5.4.5 Comparison to Human Performance

To compare the capabilities of CIMA-Pose with human performance, model predictions are obtained on the inter-rater frames of the CIMA recordings and evaluated with respect to the statistics of human annotators. Table 5.14 presents the performance of CIMA-Pose and human annotators, summarizing the error across all keypoints, both with respect to the number of pixels in a 360 × 640 sized image and relative to the size of the nose-chest segment. The different measurements are denoted as Keypoint Error and Relative Keypoint Error respectively. The Relative Keypoint Error is a way to evaluate precision by relating to the PCK_i metrics. By displaying that the 95% quantile of human annotators is at 14.60% (Table 5.14), we estimate that human performance on PCK_i@0.15 would be approximately 95. Similarly, we consider the mean of 2.13 for Keypoint Error with human annotators to be a good estimate for human precision on MKE. Observing that CIMA-Pose achieves a 95% quantile of 20.06% on Relative Keypoint Error and a mean of 3.06 for Keypoint Error, human annotations are more accurate than predictions of CIMA-Pose. Figure 5.10 compares the performance of CIMA-Pose and human annotators in a box plot displaying error distributions across the seven body parts.

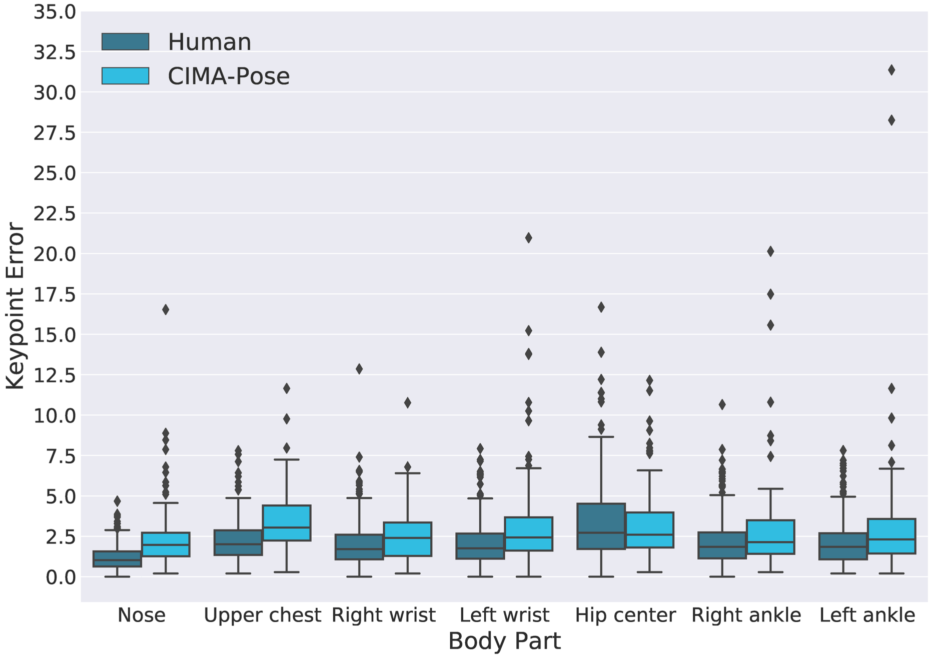


Figure 5.10: Comparison of Keypoint Error between CIMA-Pose and human annotators

5.4.6 Visual Inspection

To verify the performance of the alternative methods, body part tracking is performed with a recording displaying an infant for 3 minutes. The video was recorded using the standardized set-up but was not included in the developed CIMA Keypoints Dataset. Applying the different models from our experiments, the following set of videos are supplied with the thesis:

- *BaselineI.mp4*: Off-the-shelf CMU-Pose model
- *BaselineII.mp4*: CMU-Pose model fine-tuned on CIMA data
- *BaselineIII.mp4*: CMU-Pose model trained from scratch
- *CMU-Pose_Light.mp4*: CMU-Pose Light
- *CIMA-Pose_Raw.mp4*: Raw CIMA-Pose predictions
- *CIMA-Pose_MovingAverage.mp4*: CIMA-Pose with smoothing

In addition, Figures 5.11 to 5.14 include visualizations of CIMA-Pose predictions on images from the test portion of the CIMA Keypoints Dataset. The first three figures contain images where all keypoints were predicted within the 15%, 30% and 50% PCK_i thresholds respectively. In Figure 5.14, all images contain at least one keypoint with a prediction outside the 50% threshold. The obtained keypoints are displayed together with body part connections in colors, while the ground truth keypoints and connections are shown in white.

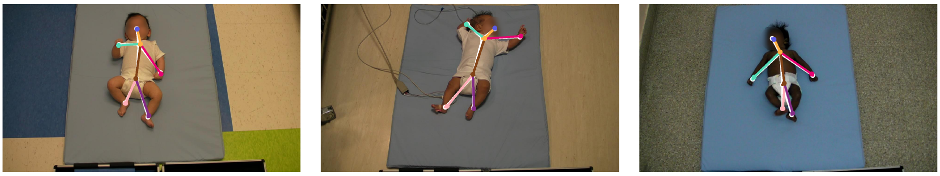


Figure 5.11: Images with all keypoints predicted within the 15% threshold from the annotated keypoint locations

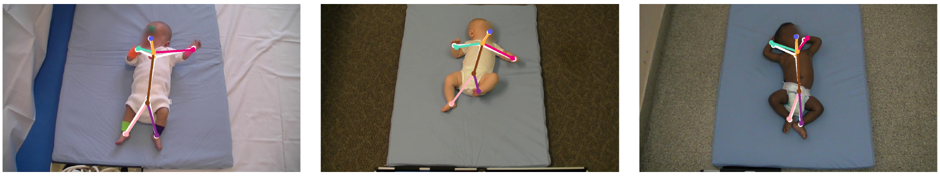


Figure 5.12: All predicted keypoints reside within the 30% threshold

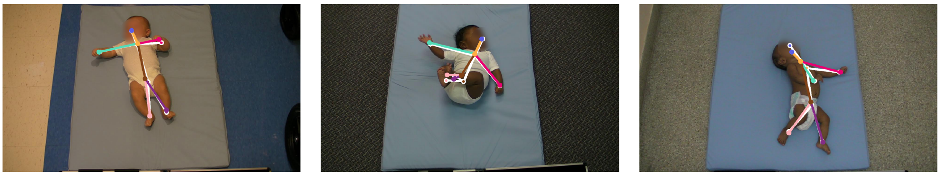


Figure 5.13: All keypoints are within the 50% threshold

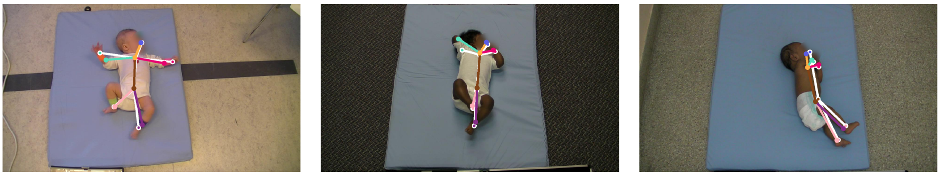


Figure 5.14: Predicted poses containing minimum one keypoint placed outside the 50% threshold

Besides inspecting the model performance on images contained in the CIMA Test set, CIMA-Pose is evaluated on a selection of infant recordings which vary from the videos recorded with the set-up discussed in Section 4.2.1. In particular, these videos differ from the standard recordings by utilizing smartphone cameras, being recorded in more noisy environments and that the infants displayed are of different scales and not necessarily placed in the center of the image. The examples of Figures 5.15 and 5.16 outline both the strengths and weaknesses of the model when it comes to predicting body part locations in more general circumstances.

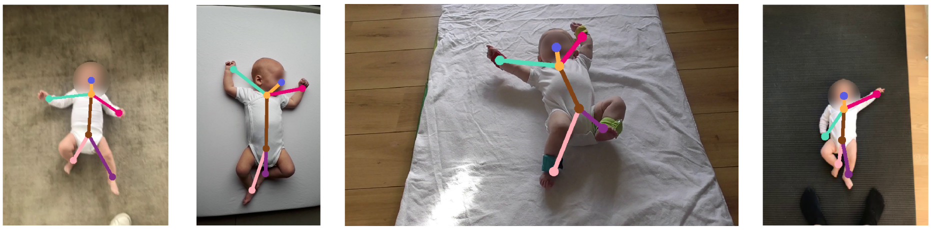


Figure 5.15: The ability of CIMA-Pose to adapt to more general circumstances

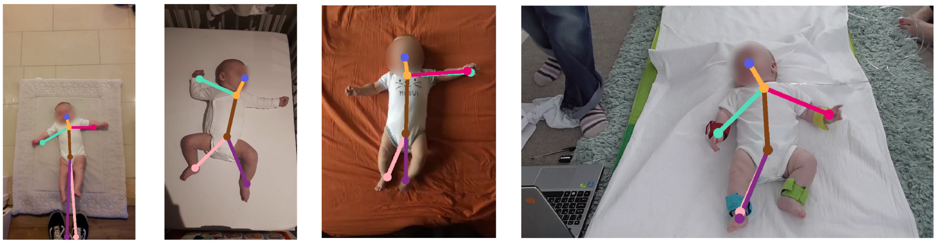


Figure 5.16: Examples displaying how the model might be challenged

5.4.7 Expert Assessment

To evaluate the quality of our body part tracker, a GMA expect provided scores from 1 to 5 with respect to the ability to follow body parts in video recordings of infants. A summary of the results is displayed in Table 5.15. From this, an average score of 4.78/5 is obtained. Additionally, the GMA expert made the following statement: *"I consider the tracking of body parts to work very well, irrespective of lighting conditions, people moving in the surroundings, cables connected to the child, clothing, jewelry, skin color, quick and complex movements"*.

In the cases where the observer gave a rating of 4, it was explained that one or two keypoints slipped for a brief moment during the 30 seconds.

Table 5.15: Qualitative evaluation of body part tracking

	Number of videos	**Fraction of videos**
(1) Strongly disagree	0	0.00%
(2) Disagree	0	0.00%
(3) Neither disagree nor agree	0	0.00%
(4) Agree	9	22.50%
(5) Strongly agree	31	77.50%

Chapter 6

Discussion

Based on the results being presented and explained in Chapter 5, the upcoming chapter is concerned with interpreting the results and to draw general reflections from these.

In Section 6.1, the self-developed CIMA Keypoints Dataset is assessed both with respect to what it contains and based on the quality and consistency of the annotation work. Subsequently, Section 6.2 discusses the obtained baseline models, before interpreting the implications of the process of model exploration in Section 6.3. Section 6.4 assesses the CIMA-Pose model with respect to precision and efficiency, and consider the method with respect to the tracking procedure and potential for clinical use. Finally, in Section 6.5, the research questions are brought up and answered based on the findings of the project.

6.1 CIMA Keypoints Dataset

In this project, the developed dataset has played a crucial role in facilitating enhanced performance on the task of localizing and tracking body parts in infant recordings. This is made clear by the improved precision of Baseline II compared to the off-the-shelf model of Baseline I (Table 5.2).

With regard to the quality of the dataset, evaluations of the annotation consistency displayed that there were small differences across annotators. However, the observed variance highlights the difficulty of the task at hand. In particular, the larger variations on some body parts might indicate some ambiguity in the definitions. For instance, the hip center seems to be more of an ambiguous point compared to the nose tip. In addition, this might be a result of difficulties in annotating the hip center in situations where the body is rotated or when unusual poses occur.

Because of models being pre-trained on more general datasets, the main objective of the CIMA Keypoints Dataset was to cover the large variety of different infant poses as good as possible. By consisting of random frames from a large set of CIMA recordings, a large selection of different poses are represented, with most of the frequently occurring body part configurations being represented comprehensively. However, the dataset might be balanced towards normal poses which could make it harder for models to learn patterns specific to poses appearing more rarely. Hence, we could have put more emphasis on constructing a dataset that contains a more balanced distribution of poses. Nevertheless, it seems like the dataset neither is too small nor too limited in variation. This presumption is based on the observation that the models are converging during the training process (Figure 5.8), and at the same time, as displayed by Figures 5.4 and 5.7, the information learned during training can be successfully applied to validation data. Overall, this indicates that the CIMA Keypoints Dataset provides information sufficient to learn general patterns relevant to the task of Infant Body Pose Estimation.

6.2 Baseline

As described in Section 4.3, we used three variations of the CMU-Pose architecture to comprise a set of baseline models. Baseline I as an off-the-shelf solution, Baseline II to take advantage of fine-tuning on the developed CIMA Keypoints Dataset, and Baseline III as an example of a model trained from scratch. The improved performance seen in Baseline II and Baseline III validated that our training procedure was successful, and confirmed that training on CIMA Keypoints data would be beneficial. The uncommon pattern observed in learning curves, with the training loss being higher than the validation loss, might be a result of the data augmentation applied during training. The rotating, shifting, and scaling of images might lead to more challenging cases, resulting in higher loss during this phase. Nevertheless, the flat tail observed in the end of the learning curves indicated that the learning process had converged.

The slight improvement in run-time efficiency of Baseline II and Baseline III reflected the reduced number of keypoints predicted compared to Baseline I. Despite being efficient in comparison to other approaches of the field, we realized that applying the models to videos is time-consuming. Hence, this was an aspect that we wanted to further optimize.

From these experiments, we concluded that training on the self-developed CIMA Keypoints Dataset would be crucial to reach high precision. The obtained precision and run-time efficiency were used for comparison at later stages of our project.

6.3 Model Exploration

Aiming to find a more efficient ConvNet architecture, Section 5.3 compared the proposed models with Baseline III. The first experiment evaluated the utilization of a lighter feature extractor. We found that the model utilizing the MobileNet architecture did not compete with the level of precision provided by the baseline models. This shows that, while MobileNet performs well on image classification, the architecture might not be sufficiently complex to handle our task. With DenseNet, the precision was closer to that of the network utilizing VGG-19, while also improving the run-time performance from 20 FPS to 24.8 FPS. To underline the differences in terms of network complexity, it is worth noting that the DenseNet feature extractor contains only 1.4 million parameters compared to the 5.9 million parameters of VGG-19. Therefore, in order to compose an efficient model, we reasoned that DenseNet would be a sensible choice despite a small drop in precision.

The experiments conducted with confidence maps as the only output, proved that this would be sufficient for achieving high precision. This suggested that the second branch is less important in single-person Human Pose estimation, as we conjectured. Based on this, we drew the conclusion that for our purpose there is no need for part affinity fields.

Considering the question of how many detection passes are really necessary, we found that most of the precision gain is obtained during the first two passes of detection. As we expected, the run-time efficiency increased with fewer passes, yielding 40.7 FPS using two detection passes. Compromising between precision and latency, we therefore preferred the model performing detection over only two rounds.

The experiments with dilated kernels gave less clear answers. As we expected, the run-time performance improved, but the modification affected the precision metrics in a negative way. However, the large improvements in run-time performance might justify the slightly reduced precision. From this experiment, we learned that a dilation rate of 2 might not fully utilize the power of dilated kernels, and that larger dilation rates should be explored.

CMU-Pose Light constituted the composition of the most promising changes discussed earlier in this section. Having a particular focus on run-time performance, this model obtained 112.4 FPS. In spite of this, a model precision close to the baseline models was achieved. Overall, the experiment indicated that the ideas implemented in CMU-Pose Light were reasonable for detecting body parts in an efficient manner, but that additional considerations had to be taken in order to solve the task with higher precision.

6.4 Validation of CIMA-Pose

6.4.1 Precision

With regard to precision, we found that the CIMA-Pose model outperforms the baselines and earlier models. In particular, compared to CMU-Pose Light, all precision metrics were improved. Compared to the baseline models, the MKE declined, showing that predictions on average were closer to the ground truth locations. In addition to this, the biggest improvement was seen in terms of PCK_i@0.15, indicating that a larger fraction of the predictions is very close to the targets. The detection stage delivers high-quality predictions in the low-resolution space of the confidence maps, while the upscaling stage ensures that the output is of sufficient precision. With this in mind, and considering that all models are trained in a similar fashion, it becomes clear that the CIMA-Pose model handles the task of Infant Body Part Estimation better than existing approaches.

Interesting to notice, CIMA-Pose improves precision compared to Baseline III even without applying upscaling. This indicates that the detector stage of CIMA-Pose in itself is very powerful. Compared to our early experiments with dilated convolutions, the important change in CIMA-Pose was the use of higher dilation rates, resulting in a larger receptive field than normal convolutions of the same kernel size. Specifically, the dilated convolutional layer of CIMA-Pose with a dilation rate of 4 would have to be replaced by a 9×9 standard convolutional layer, in order to extract the same amount of global context from the input feature maps. This simple change would significantly slow down the inference speed of the network.

In order to validate if the CIMA-Pose model was trained appropriately, we observed the learning curves of the model with respect to the training and validation portions of the CIMA Keypoints Dataset. First of all, we learned that CIMA-Pose reached a low validation loss compared to the other models. This reflects the ability of CIMA-Pose to extract information that is particularly relevant to the task of Infant Body Pose Estimation. Overall, the CIMA-Pose model displayed similar learning curves as the baseline models, with the training loss being slightly higher than the validation loss. In spite of this, the accuracy of the model on the training set is a bit higher than that of CIMA Val and CIMA Test (Table 5.10). However, the low MKE values across all sets and the similar accuracy values of the validation and test sets indicates a well-trained model.

CIMA-Pose improved estimates for the locations of all seven body parts, yet we observed that some body parts were localized more precisely than others. This could be reflected by the earlier assessed differences in variability across body parts in the CIMA Keypoints Dataset.

As illustrated by various model predictions (Figures 5.11 to 5.14), CIMA-Pose performs at a very high level in general, both with respect to different poses and by variations in lighting conditions, clothing, and skin color. The examples also indicate the challenging nature of the accuracy measures, with all thresholds being of seemingly high precision. The stability of the model was supported by performance on images from more general situations (Figures 5.15 and 5.16).

In spite of this, we have seen that there are some scenarios where the model is challenged. The most typical examples include when body parts are either not visible or when very unusual poses occur, such as when the infant turns over on the side. Additionally, infants in diapers placing a hand on the chest can result in difficulties due to low contrast. In images of more general circumstances, wrong predictions were seen to occur when body parts of other persons were close to the infant being recorded. In these examples, we also observed that there are situations where the model fail to be certain about a decision, or where a keypoint is predicted on the opposite wrist or ankle. The latter obstacle was also encountered in the standardized setting before models were trained specifically for the task of Infant Body Pose Estimation using the developed CIMA Keypoints Dataset.

Having seen this, we argue that a more general dataset might be required to perform accurate Infant Body Pose Estimation in a wider variety of circumstances. However, the model delivers precise and robust predictions in the standardized circumstances of the CIMA recordings, with only very low contrasts and rare poses challenging the model to a significant degree. Ensuring that the infants have clothing contrasting the skin color will probably eliminate the former, while constructing a dataset more balanced towards the range of rare poses might reduce the problem of the latter.

In many Machine Learning problems, human performance can be considered an upper limit for how good a model can become. As our model is trained on data which are manually annotated by humans, it is reasonable to assume that it is not possible to be more precise than the precision of the annotations themselves.

Based on the images and annotations of the inter-rater samples, human performance was compared to the capabilities of the CIMA-Pose model. Although humans on average were more precise, we observed that with respect to the keypoint of the hip center, the CIMA-Pose model displays more consistency. This indicates that the model has generalized upon the knowledge supplied through human annotations. Nevertheless, in general, we conclude that our model has not fully accomplished human performance. From this, we suggest that even higher precision can be obtained by training on more annotated samples.

Overall, CIMA-Pose displays consistent performance in the task of localizing body parts, outperforming both state-of-the-art approaches as well as the other models proposed during this project. With regard to the obtained MKE value of 3.81, in average the model is able to place a keypoint within a radius of less than 1 centimeter from the annotated location. This outlines the remarkable detection quality of CIMA-Pose.

6.4.2 Efficiency

Although high detection quality was the major aim of this project, the clinicians also wanted a solution which was fast enough for daily use in clinical practice. Having seen the baseline performance of 20 FPS, and the possible gains of a more efficient model in CMU-Pose Light, we knew that high efficiency was possible. The final CIMA-Pose model further improved from this, reaching 120.8 FPS.

Illustrated by the reduction of parameters from 52.3 million to 2.3 million, it is evident that the choices made during the process have facilitated a light ConvNet to be constructed. The proposed detection stage was crucial for this achievement. This included taking full advantage of dilated convolutions, having only two detection passes and producing a single type of output. Moreover, the utilization of DenseNet as feature extractor and the low number of detection passes played an important role in building a more efficient model.

Despite being efficient as it is, we saw that the run-time performance of CIMA-Pose could be further improved. With respect to the number of detection passes, it is possible to construct a 1-pass detection stage to obtain 125 FPS on the cost of a small precision decrease (5.12). That being said, two detection passes seems like a reasonable choice by observing that this ConvNet actually performs better than a similar model with three rounds of detection. Another modification that might be desired in situations where run-time performance is appreciated over top detection quality is reducing the number of transposed convolutions in the upscaling stage. Table 5.13 displays how a single upsampling layer can be utilized to reach MKE of 4.09 while operating at a speed of 133.3 FPS. This shows that the CIMA-Pose model offers flexibility in the trade-off between precision and efficiency.

6.4.3 Tracking

With regard to tracking of body parts in time, performing initial predictions from a single frame has proven to both have strengths and weaknesses. On the negative side, in situations where a body part is not visible, like when the arm is placed behind the neck, the model is often not precise in its predictions. These are situations which often last for more than a short while, and hence might not be easily overcome using the simple Moving Average algorithm. Therefore, we suggest that being presented with predictions of preceding frames could help the model to improve predictions in these cases. However, such an approach would increase the model complexity, and negatively affect the run-time efficiency.

On the positive side, the single-frame approach lets the model quickly rediscover a body part once it becomes visible, without being affected by recent uncertain predictions. Additionally, using the Moving Average algorithm, small fluctuations of model predictions are removed, which slightly improves the visual impression.

Having seen that certain videos contained short spikes resulting from a single misplaced body part prediction, we discussed whether such cases could be avoided by considering unnaturally quick movements as outliers. We concluded that this is probably achievable, but did not find the time to evaluate such a process as part of this project.

6.4.4 Expert Assessment and Clinical Use

The positive feedback received from the qualitative expert evaluation serves as a great indication that body part tracking provided by CIMA-Pose is sufficiently precise to perform this task in clinical use. In particular, the statement made about robustness in varied conditions indicates that our method has reached a state of high stability.

6.5 Answering Research Questions

As stated by the two research questions, this project was conducted to evaluate whether body part tracking of infants in videos could be implemented in a way that was both accurate and efficient. While we have seen that there are a great number of challenges to solve this completely, we argue that the obtained results demonstrate great capabilities for the CIMA-Pose model. Following, we will discuss the findings in light of each research question.

1. Can a Deep Learning model solve the task of Infant Body Pose Estimation accurately?

Starting with the pre-trained CMU-Pose model (Baseline I), our hypothesis that this could be solved using Deep Learning was strengthened, reflected by the high level of coarse accuracy, with PCK_i@0.5 of 97.1. This proved that the model in most cases was able to locate the correct image region for each of the selected body parts. However, we observed much lower values in the fine-grained specter addressed by PCK_i@0.3 and PCK_i@0.15, which resulted in less accurate tracking, with small movements being lost. In addition, visual inspection showed that unusual postures confused the model, which for instance led to predictions of the left ankle being on the right ankle, and vice versa.

With Baseline II and III, we displayed that training on the annotated images of infants reduced the precision issues seen in Baseline I, with PCK_i@0.3 of 98.3. However, we expected that even more improvement could be made, especially with respect to the PCK_i@0.15 metric. This was addressed in the proposed CIMA-Pose model, with dilated convolutions and upscaling as important contributions, reaching PCK_i@0.15 of 82.2, compared to 73.5 for the best performing baseline model.

Given these results, and the mean score of 4.78/5 on the qualitative assessment, we are confident in confirming that the model sufficiently addressed this research question.

2. Can this be done efficiently such that the model can be integrated in a system for medical diagnosis?

As the solution previously used by the CIMA group involved both costly computation and time-consuming manual labor, the second goal of our project was to find a solution omitting these challenges.

Using CMU-Pose in the baseline models, we observed that a run-time efficiency of 20 frames per second was achievable, with relatively low keypoint error and high accuracy. In the experiments, we displayed that using a more efficient feature extractor, dilated convolutions, fewer passes and a single output branch drastically increased the efficiency, leading to CMU-Pose Light achieving 112.4 frames per second. However, this increase in run-time performance came at the cost of reduced model precision.

With the final model of CIMA-Pose, we managed to obtain a run-time efficiency of 120.8 frames per second, while maintaining high precision. In real-life use, this means that a four-minute video, of 25 frames per second, could be processed in less than one minute. The great improvement from CMU-Pose will save precious time in clinical use. Compared to the cumbersome and expensive process of manual labor in the current solution, we conclude that CIMA-Pose represents an important contribution within the medical domain as well in the field of Deep Learning concerning the efficiency aspect.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have shown the importance of developing a good method for body part tracking. We have presented current methods to diagnose CP, and how these methods require potentially harmful use of anesthesia. It has been shown how clinical experts are able to predict CP based on GMA using Gestalt Perception, but that the dependency of manual work motivated an automated solution. In line with this, we have presented background theory of Computer Vision and Deep Learning, and how subfields such as Image Segmentation or Human Pose Estimation could be applied to our task. Inspired by state-of-the-art methods and development in the field of Deep Learning, we have proposed the CIMA-Pose model, that is able to outperform currently available approaches both in terms of precision and run-time efficiency.

With this in mind, we have shown that a Deep Learning-based model for localizing body parts can be used to track movements facilitating the creation of an automated solution for CP detection. Although body tracking is only one component in such a solution, it is essential to provide accurate data from this step to the other components. As validated by a GMA expert, the method has proven to sufficiently follow the movements of infants in CIMA recordings irrespective of lighting conditions, clothing, skin color, and movements of other individuals in the surroundings. Great improvement in run-time efficiency has also shown that this approach can be efficiently applied in daily clinical practice, without the use of supercomputers. However, we have also seen some limitations with this method, in particular when applied to images not following the standardized set-up of the CIMA recordings.

Nevertheless, we can confidently conclude that both our research questions have been resolved. The supplied video examples have also demonstrated how infant movements are tracked precisely by CIMA-Pose. Overall, our evaluation shows the practicality and feasibility of the proposed approach.

7.2 Future Work

With respect to the aspects earlier discussed, we suggest three natural paths that could be explored further.

Dataset

To make the model more stable on videos other than the standardized CIMA recordings, a more diverse dataset should be developed and utilized to train the model further. The dataset could also be increased in size, to evaluate whether this will further improve model precision. A similar annotation process as followed in this project could easily be extended to take this into account, by collecting and annotating data from relevant video recordings.

Intelligent Tracking

By performing model predictions using information from more than one frame, precision could be improved. This could be more evident in challenging situations, having the opportunity to account for predictions of body part locations of preceding frames in a more intelligent way.

Detection of CP

While body part tracking provides the basis for computer-based GMA, a system which analyzes the movements is needed to perform CP detection. In such a system, the CIMA-Pose model is essential, by supplying movement information, which will be utilized in the subsequent stage to perform movement analysis. The capabilities of this component in detection of CP can be considered a final validation of whether CIMA-Pose sufficiently captures the infant movements, and whether it is possible to automate the whole process of detecting CP. Before applying the computer-based system in clinical practice, all components should naturally be thoroughly evaluated through quality analysis and risk assessment.

References

[1] Cao Z, Simon T, Wei SE, Sheikh Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. 2016 nov;Available from: http://arxiv.org/abs/1611.08050.

[2] Huang G, Liu Z, Weinberger KQ. Densely Connected Convolutional Networks. CoRR. 2016;abs/1608.06993. Available from: http://arxiv.org/abs/1608.06993.

[3] Rosenbaum P, Paneth N, Leviton A, Goldstein M, Bax M, Damiano D, et al. A report: The definition and classification of cerebral palsy April 2006. Developmental Medicine and Child Neurology. 2007;49(SUPPL.109):8–14.

[4] Prechtl HFR. Qualitative changes of spontaneous movements in fetus and preterm infant are a marker of neurological dysfunction. Early Human Development. 1990;23(3):151 – 158. New Studies on Movement Assessment in Fetuses and Preterm Infants. Available from: http://www.sciencedirect.com/science/article/pii/0378378290900117.

[5] Adde L. Prediction of Cerebral Palsy in Young Infants. Doctoral Theses at NTNU. 2010;50.

[6] Accardo PJ, Capute AJ. Capute & Accardo's Neurodevelopmental Disabilities in Infancy and Childhood: Neurodevelopmental diagnosis and treatment. vol. 1. Brookes Pub; 2008.

[7] Beaino G, Khoshnood B, Kaminski M, Pierrat V, Marret S, Matis J, et al. Predictors of cerebral palsy in very preterm infants: the EPIPAGE prospective population-based cohort study. Developmental Medicine & Child Neurology. 2010;52(6).

[8] Bax MC. Terminology and classification of cerebral palsy. Developmental Medicine & Child Neurology. 1964;6(3):295–297.

[9] Rosenbaum P, Paneth N, Leviton A, Goldstein M, Bax M, Damiano D, et al. A report: the definition and classification of cerebral palsy April 2006. Dev Med Child Neurol Suppl. 2007;109(suppl 109):8–14.

[10] Hadders-Algra M. General movements: a window for early identification of children at high risk for developmental disorders. The Journal of pediatrics. 2004;145(2):S12–S18.

[11] O'Shea TM. Diagnosis, treatment, and prevention of cerebral palsy in near-term/term infants. Clinical obstetrics and gynecology. 2008;51(4):816.

[12] Einspieler C, Peharz R, Marschik PB. Fidgety movements – tiny in appearance, but huge in impact. Jornal de Pediatria. 2016;92(3, Supplement 1):S64 – S70. Available from: http://www.sciencedirect.com/science/article/pii/S0021755716000516.

[13] Bosanquet M, Copeland L, Ware R, Boyd R. A systematic review of tests to predict cerebral palsy in young children. Developmental Medicine Child Neurology. 2013;55(5):418–426. Available from: http://dx.doi.org/10.1111/dmcn.12140.

[14] Ferrari F, Einspieler C, Prechtl H, Bos A, Cioni G. Prechtl's method on the qualitative assessment of general movements in preterm, term and young infants. Mac Keith Press; 2004.

[15] Hadders-Algra M. The assessment of general movements is a valuable technique for the detection of brain dysfunction in young infants. A review. Acta Paediatrica. 1996;85(s416):39–43.

[16] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016. http://www.deeplearningbook.org.

[17] Chen CHCh. Handbook of pattern recognition and computer vision. World Scientific; 2016.

[18] Nielsen MA. Neural networks and deep learning. URL: http://neuralnetworksanddeeplearningcom/(visited: 01112016). 2017;Available from: http://neuralnetworksanddeeplearning.com/index.html.

[19] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–1105.

[20] Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. CoRR. 2015;abs/1511.07122. Available from: http://arxiv.org/abs/1511.07122.

[21] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? 2014 nov;Available from: http://arxiv.org/abs/1411.1792.

[22] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09; 2009. .

[23] Otsu N. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics. 1979;9(1):62–66.

[24] Ikonomatakis N, Plataniotis K, Zervakis M, Venetsanopoulos A. Region growing and region merging image segmentation. In: Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on. vol. 1. IEEE; 1997. p. 299–302.

[25] Horowitz SL, Pavlidis T. Picture segmentation by a tree traversal algorithm. Journal of the ACM (JACM). 1976;23(2):368–388.

[26] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. Berkeley, Calif.: University of California Press; 1967. p. 281–297. Available from: https://projecteuclid.org/euclid.bsmsp/1200512992.

[27] Sigal L. Human pose estimation. In: Computer Vision. Springer; 2014. p. 362–370.

[28] Elhayek A, Aguiar E, Jain A, Tompson J, Pishchulin L, Andriluka M, et al. Efficient ConvNet-based Marker-less Motion Capture in General Scenes with a Low Number of Cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. .

[29] Pishchulin L, Andriluka M, Schiele B. Fine-grained Activity Recognition with Holistic and Pose based Features. CoRR. 2014;abs/1406.1881. Available from: http://arxiv.org/abs/1406.1881.

[30] Adde L, Helbostad JL, Jensenius AR, Taraldsen G, Støen R. Using computer-based video analysis in the study of fidgety movements. Early Human Development. 2009;85(9):541 – 547. Available from: http://www.sciencedirect.com/science/article/pii/S0378378209000814.

[31] Adde L, Helbostad JL, Jensenius AR, Taraldsen G, Grunewaldt KH, Støen R. Early prediction of cerebral palsy by computer-based video analysis of general movements: a feasibility study. Developmental Medicine & Child Neurology. 2010;52(8):773–778.

[32] Stahl A, Schellewald C, Stavdahl Ø, Aamo OM, Adde L, Kirkerod H. An optical flow-based method to predict infantile cerebral palsy. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2012;20(4):605–614.

[33] Rahmati H, Martens H, Aamo OM, Stavdahl Ø, Støen R, Adde L. Frequency analysis and feature reduction method for prediction of cerebral palsy in young infants. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2016;24(11):1225–1234.

[34] Rahmati H, Dragon R, Aamo OM, Van Gool L, Adde L. Motion segmentation with weak labeling priors. In: German Conference on Pattern Recognition. Springer; 2014. p. 159–171.

[35] Rahmati H, Aamo OM, Stavdahl Ø, Dragon R, Adde L. Video-based early cerebral palsy prediction using motion segmentation. Conf Proc IEEE Eng Med Biol Soc. 2014;2014:3779–3783.

[36] Hesse N, Stachowiak G, Breuer T, Arens M. Estimating body pose of infants in depth images using random ferns. In: Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on. IEEE; 2015. p. 427–435.

[37] Hesse N, Schröder AS, Müller-Felber W, Bodensteiner C, Arens M, Hofmann UG. Body pose estimation in depth images for infant motion analysis. In: Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE. IEEE; 2017. p. 1909–1912.

[38] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 07-12-June-2015; 2015. p. 3431–3440.

[39] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39(12):2481–2495.

[40] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 2017-October; 2017. p. 2980–2988.

[41] Ren S, He K, Girshick R, Sun J. Faster R-CNN; 2017. Available from: http://arxiv.org/pdf/1506.01497v2.pdf.

[42] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV). 2015;115(3):211–252.

[43] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR. 2014;abs/1409.1556. Available from: http://arxiv.org/abs/1409.1556.

[44] Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, et al. Going Deeper with Convolutions. CoRR. 2014;abs/1409.4842. Available from: http://arxiv.org/abs/1409.4842.

[45] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. CoRR. 2015;abs/1512.03385. Available from: http://arxiv.org/abs/1512.03385.

[46] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR. 2017;abs/1704.04861. Available from: http://arxiv.org/abs/1704.04861.

[47] Toshev A, Szegedy C. DeepPose: Human pose estimation via deep neural networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2014;p. 1653—-1660.

[48] Wei SE, Ramakrishna V, Kanade T, Sheikh Y. Convolutional Pose Machines. 2016 jan;Available from: http://arxiv.org/abs/1602.00134.

[49] Newell A, Yang K, Deng J. Stacked Hourglass Networks for Human Pose Estimation. 2016 mar;Available from: http://arxiv.org/abs/1603.06937.

[50] Yang W, Li S, Ouyang W, Li H, Wang X. Learning Feature Pyramids for Human Pose Estimation. 2017 aug;Available from: http://arxiv.org/abs/1708.01101.

[51] Güler RA, Neverova N, Kokkinos I. DensePose: Dense Human Pose Estimation In The Wild. 2018 feb;Available from: http://arxiv.org/abs/1802.00434.

[52] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common objects in context. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 8693 LNCS; 2014. p. 740–755.

[53] Insafutdinov E, Andriluka M, Pishchulin L, Tang S, Levinkov E, Andres B, et al. ArtTrack: Articulated Multi-person Tracking in the Wild. 2016 dec;Available from: http://arxiv.org/abs/1612.01465.

[54] Iqbal U, Milan A, Gall J. PoseTrack: Joint Multi-Person Pose Estimation and Tracking. 2016 nov;Available from: http://arxiv.org/abs/1611.07727.

[55] Andriluka M, Pishchulin L, Gehler P, Schiele B;. .

[56] Andriluka M, Iqbal U, Milan A, Insafutdinov E, Pishchulin L, Gall J, et al. PoseTrack: A Benchmark for Human Pose Estimation and Tracking. CoRR. 2017;abs/1710.10000. Available from: http://arxiv.org/abs/1710.10000.

[57] Johnson S, Everingham M. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In: Proceedings of the British Machine Vision Conference; 2010. Doi:10.5244/C.24.12.

[58] Shipman JW. Tkinter 8.5 reference: a GUI for Python;. Available from: http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html.

[59] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely Connected Convolutional Networks. 2018;Available from: https://arxiv.org/pdf/1608.06993.pdf.

[60] Yang W, Li S, Ouyang W, Li H, Wang X. Learning Feature Pyramids for Human Pose Estimation. 2017 8;Available from: http://arxiv.org/abs/1708.01101.

[61] Chollet F, et al.. Keras; 2015. https://keras.io.

[62] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Software available from tensorflow.org. Available from: https://www.tensorflow.org/.

[63] Yang Y, Ramanan D. Articulated human detection with flexible mixtures of parts. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013;35(12):2878–2890.