IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

3D Infant Pose Estimation from RGB Video using Deep Learning

Author:
Simon Ellershaw

Supervisors:
Luca Schmidtke
Dr Bernhard Kainz

Submitted in partial fulfillment of the requirements for the MSc degree in MSc Computing Science of Imperial College London

June 2020

Abstract

The current state of the art methods for 3D adult pose estimation from RGB video are based on modern deep learning techniques including convolutional neural networks. However, as yet, similar methodologies have not been transferred to the infant domain where classical or empirically adapted adult pose estimation models top performance on the benchmark synthetic infant MINI-RGBD dataset. This project developed a deep learning 3D infant pose estimation model using RGB video input. This has a 2 stage design consisting of a 2D infant pose estimation network and a 3D lifting network adapted from the adult domain via transfer learning.

The 2D and 3D models achieved a PCKh 1.0 value, analogous to accuracy, of 93.77% and 65.20% respectively. This represents a 7.23% and 14.11% increase in performance over the current state of the art models on the same data from the MINI-RGBD benchmark dataset. Moreover, unlike the current SOTA, this methodology does not require a depth channel input.

The development of such a model has implications for the advancement of several cutting edge computer vision applications. The primary example and underlying motivation for this work is the development of an automated general movement assessment for infants, for which the current bottlenecked is the lack of an accurate and easily deployable infant pose estimation model. Such a system has the potential to revolutionise the diagnosis of a range of disorders, including cerebral palsy, for which an infant's abnormal movements are symptomatic.

Acknowledgements

I would like to firstly thank my supervisors, Luca Schmidtke and Dr Bernhard Kainz, for the time and effort they have dedicated to steering me through this project. Without their help and guidance, the work presented in this report may have not progressed far beyond the PyTorch download process.

Secondly, I would like to thank my parents for at least feigning interest in my work over the last couple of months and for the endless cups of coffee during lockdown.

Contents

Chapter 1

Introduction

The current state of the art (SOTA) approaches to infant pose estimation use classical methods [1] or empirically adapt deep learning models trained on adult datasets [2]. This is in contrast to the adult domain in which deep learning approaches have been the SOTA since 2014 [3] and has led to a leap in performance on benchmark datasets [4].

One reason for this lack of progression is that currently, no large infant dataset exists on which to train a modern deep learning model. However, the use of transfer learning to first train a model on the large adult datasets [5, 4, 6] that are available and then finetune on the smaller infant datasets [7] may provide a solution.

Evidence that such an approach could be successful has been shown by the, as yet unpublished work, of the Movement Assessment for Very Early Health Assessment (MAVEHA) project currently being carried out at the BioMedIA lab at Imperial College London. They have successfully developed a 2D infant pose estimation model based on a baseline deep learning model proposed by Xiao et al. [8] which was finetuned on an infant dataset through the use of transfer learning.

This project aims to build upon this work and so extend the scientific literature by investigating if deep learning techniques developed in the 3D adult pose estimation domain could be successfully adapted to the problem of 3D infant pose estimation via transfer learning. This will be done by developing a baseline 3D infant pose estimation model which can then be tested on the infant MINI-RGBD [7] dataset and compared to the current SOTA approaches [1, 2].

The development of such a model is not a trivial application of a known technology. However if successful, the primary application of this work would be the potential development of an automated diagnostic tool for a range of disorders, including cerebral palsy, which can be linked to an infant's abnormal movement patterns. Such a system would currently require a bulky 3D tracking system, such as MOCAP [9], which is impractical in a clinical environment with uncooperative neonates. Data for a model requiring just an RGB video input though could be recorded using equipment as ubiquitous as a smartphone.

Beyond the clinical use of such an infant pose estimation model, this project contributes to the wider research effort to develop pose estimation models that generalise well when deployed in the wild. Such models are being developed for a range of domains such as autonomous vehicles [10] and virtual dressing mirrors [11]. Furthermore, this work will extend the knowledge base regarding the methodologies required to perform out of domain adaptation of existing deep learning models. This will increase the generality and so use cases for previously proposed models.

This report outlines in detail the relevant background and related work to the project, the methodology developed to implement the 3D infant pose estimation model, the resulting performance of the model and an evaluation of the model's achievements and limitations.

Chapter 2

Background

This chapter outlines the relevant previous work and background associated to this project including the medical background, previous approaches to infant and adult pose estimation, the current state of the MAVEHA project and the ethical considerations of this work.

2.1 Medical Background

Studies have shown that qualitative movement analysis of infants can potentially be used to diagnose a range of conditions such as autism [12] and neurological dysfunction [13]. The most widely cited example is its use as an indicator of Cerebral Palsy (CP) [14]. This condition affects approximately 0.2% of all live births and is caused by abnormal development or damage to the brain. Symptoms can include abnormal muscle tone, spasticity and impaired motor skills [15].

Early interventions, before the age of 6 months, have been shown to have a positive effect on the motor and cognitive abilities of CP patients in later life [16]. These take the form of coaching the child's caregiver on areas such as the infants' sensorimotor development, self-regulation, early communication skills and attention [17, 18]. However, on average diagnosis is made and communicated to the affected child's family at 11 months and for less severely affected children this period can be as long as 2 years [19].

The current recommended approach to clinicians when diagnosing patients at a high risk of CP is to use clinical history in conjunction with the available standardised clinical tools. Neurological examinations, such as MRI scans, can be used but have been shown to have a large variation in predicting developmental outcome [20] and reliable diagnosis using this technique usually only occurs at 1 to 2 years [16].

An alternative technique is based on the work of Prechtl [13]. He analysed the spontaneous movements of infants, known as general movements (GMs). These GMs emerge at a fetal stage then disappear at around 4-5 months when goal-directed motor behaviour emerges. Developmental transformations of the nervous system result in changes to the typical form of GMs over time. These can be grouped into phases characterised by their complexity and variation. The final phase occurs at 2 to 5 months at which point GMs are characterised as having a fidgety quality. This fidgety age has been found as the optimal point to identify atypical GMs, characterised by a reduction in their complexity [21].

To detect infants with CP, a video-based Qualitative Assessments of General Movements (GMA) was developed by first visually grading movement complexity and then assessing the fidgety movement of the infant. This grading is done on a scale from normal optimal to definitely abnormal. Infants with abnormal movement quality have a high risk of developing CP [20]. This GMA procedure has been shown to be the most reliable CP diagnostic and predictive technique for young infants [22].

However, GMAs require trained experts who need regular practice and re-calibration to maintain reliability. Even with these measures in place, human variability still has a negative influence on the effectiveness of GMAs. This coupled with the cost of this high-level expertise impedes the application of GMAs in broad clinical practice [23].

2.1.1 Automated General Movement Assessment

A solution to the problems outlined in Section 2.1 would be an automated screening tool which would act as an alternative to the current expert dependent GMAs. To enable its widespread use in general paediatric practise the tool should be low-cost and only require commercially available equipment that does not influence the infants' movements.

The development of such a system would transform the diagnosis of conditions such as CP within the community. This is due to its ability to be deployed in a wide range of settings from hospitals to the cot-side, via a smartphone, coupled with the low levels of training that would be required to use the tool.

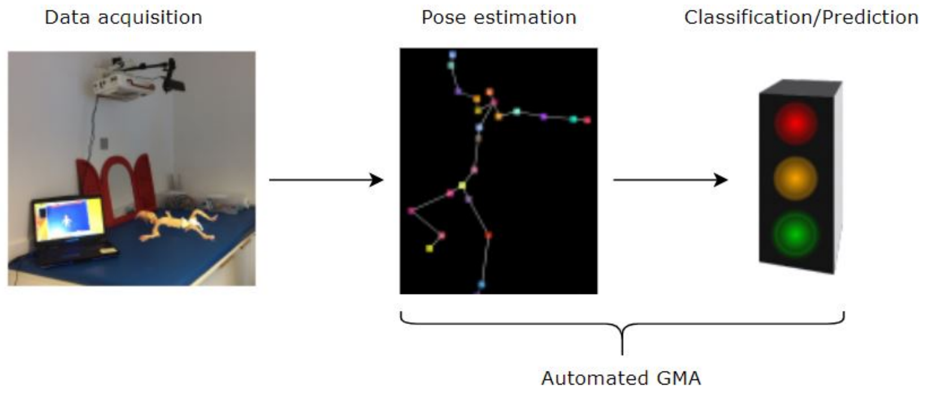

Figure 2.1: Proposed automated GMA pipeline. Data is collected then a time series of the infant's poses are extracted and used as the input to a machine learning classifier of the medical outcome [7].

The computing challenge is to take the captured data and develop an automatic analysis tool from this. In the literature, this is seen to require solutions to two distinct problems [24]. The first is to extract the motion features from the data into a continuous time series of the kinematics of the limb and trunk movements i.e. poses, either in the form of the 2D or 3D body joint positions. These time series of poses then form the input to a machine learning classifier which identifies infants with abnormal GMs and so quantitatively predicts the risk of conditions such as CP [25]. This analysis pipeline is shown visually in Fig 2.1. This project will aid the of the first stage of this pipeline by developing the techniques to enable 3D infant pose estimation.

Beyond the clinical use of such a model, this project will contribute to the wider research effort to develop pose estimation models that generalise well when deployed in the wild. Development of such algorithms in recent years [5, 26] has led to the application of pose estimation technology in domains such as autonomous vehicles [10] and virtual dressing mirrors [11]. The release of publicly available pose estimations models, such as OpenPose and Apple's augmented reality PoseNet, aims to further the adoption of the technology. However, many such libraries do not account for the differing anatomy of infants and so fail to generalise to this section of the population [27, 28]. This project will demonstrate approaches which could be used to further enrich these models.

2.2 Infant Pose Estimation Techniques

This section presents an overview of the range of approaches used to capture and extract the motion data of infants movements. It is of note that all the extraction methods outlined in this section are so-called 'classical methods'.

2.2.1 Wearable Motion Sensors

Several systems have been proposed using wearable sensors to capture an infant's movements, these include electromagnetic sensors [29, 30], wired [31] and wireless accelerometers [32] or inertial measurement units in combination with pressure mattresses [33]. Although a recent study has shown that wearable sensors do not affect the frequency of leg movement [34] many practical issues remain with the technology. For example, Karch et al. reported that two-thirds of their recordings had to be stopped to re-position sensors due to technical difficulties or the infant crying [30]. This is further reinforced by MAHEVA's difficulties of data collection using magnetic sensors, see Section 2.4.1. Furthermore, the setup and calibration of such systems are time-consuming and the equipment in some cases is specialised and expensive. For these reasons, the focus of the field and this project has shifted to vision-based approaches.

2.2.2 RGB Video Source

RGB video cameras are easy to use, cheap, require no setup or calibration, are guaranteed not to interfere with an infant's movement and in the modern smartphone era are ubiquitous. This makes them an optimal data capture tool in this context and the widespread implementation in clinical practice of an automated GMA tool using an RGB camera is easy to imagine. This is in stark contrast to the properties of the wearable motion sensor systems outlined in Section 2.2.1. However, a more sophisticated method is required to extract the pose from each frame and so gain the motion information from this data source. Outlined here is the development of these methods.

An initial vision approach was introduced in 2006 by Adde et al. [35] They looked at the motion of the infant as a whole rather than analysing individual limbs. Two consecutive frames were taken and the difference between them was found to give a so-called motion image. From this motion image, they calculated the centroid of motion, the pixel position at the centre of the motion regions in the image. Also, a motiongram was constructed by compressing all motion images in the video sequence either by summing over the rows or columns of each image. This gave a compact impression on the magnitude and location of the infants' movements. The accuracy of the system was not evaluated.

In 2012 Stahl et al. used an optical flow-based method to perform motion tracking [36]. Points were initialised and distributed across a regular grid and then tracked over time. This approach was evaluated by manually selecting 5 points at the head, feet and hands then manually correcting the tracking errors. They displayed their results graphically but no average accuracy result was published.

Rahmati et al. built upon this optical flow approach using weak supervision [37] in 2015. Body segments were initialised by manual labelling and then tracked using optical flow fields. If the trajectory was ended by occlusion or fast motion, a particle matching algorithm was used to connect a newly started trajectory to the relevant body segment. The accuracy of the system was evaluated on 20 manually annotated frames from 10 infant sequences. An F-measure of 96% was found by calculating the overlap between the estimated segmentation and the ground-truth. At the time this tracking method was found to be state of the art (SOTA) compared to other approaches [37]. Furthermore this method generalised well to other problem spaces such as the Freiburg-Berkely data set [38], containing other moving objects such as cats and ducks, where it was also found to be SOTA achieving an F-measure of 77%.

2.2.3 RGB-D Video Source

In recent years low-cost RGB-D sensors have been commercially released. The most well-known of these being the Microsoft Kinect, which was released as a game controller but due to its affordable price soon became widely used in research [39]. The Kinect was recently upgraded in 2020 with the release of the Azure Kinect, which has a specific emphasis on enabling its application to future computer vision systems rather than gaming. However, the motion tracking API provided for both devices [28], was designed for humans taller than 1 meter and so does not generalise well to infants. Research in the field has shifted to transform this RGB-D data into information on the infant's pose with the underlying assumption that the added depth channel will result in more accurate poses being extracted relative to RGB data sources.

The first of these methods of particular note was Olsen et al. which applied an existing pose estimation algorithm to infants [40]. This approach assumed that the extremities of the infant have the maximum geodesic distance to the centre of the body. The body centre is found by threshold filtering to leave only the infant's clothing. Then 5 anatomical extremities from this central point are found. These are then labelled head, left or right hand and left or right foot depending on the point's spatial location and the orientation of the path to the body centre. 3D positions of other joints, such as the knees and elbows, are then calculated by fractional distances from the body centre to the relevant extremity. This results in the extraction of the 3D positions of 11 joints. This was evaluated by comparison to manually annotated ground-truths. The AJPE was found to be 9 cm. It is of note that the highest error, of 15cm, occurred for the hand and elbow positions.

Olsen et al. built upon this methodology in a model-based approach [41]. A model infant body was constructed from basic geometric shapes such as spheres and cylinders. Firstly the size parameters of the infant were determined. Then using their previous method an initial pose was found. The body model was then fitted to the segmented infant point clouds which were computed from the RGB-D video frames. An objective function, given by the difference between the closest points from the model and point cloud, was then optimised using the Levenberg-Marquardt algorithm [42] with respect to the model parameters. They evaluated the AJPE to be 5cm and the error in the hands to be 7cm, an improvement of approximately 50% compared to their previous work.

In a modern take on the work of [35] Cenci et al. also took the difference between two frames [43]. The resultant image was then noise filtered and segmented into motion blobs using a threshold. K-means clustering was then used to assign each movement blob to one of the infant's limbs. A state vector was then calculated for each frame containing information on each limb's movement in the frame relative to its previous neighbouring frame. No evaluation was given for this method.

An alternative approach using a stereo camera was proposed by Shivakumar et al. as this system provided higher depth resolution compared to the existing RBG-D cameras [44]. The body centre was initially found by colour thresholding. To this coloured region, an ellipse was fitted and tracked. Further tracking of the torso centre, legs, hands, and head regions was performed based on their colour after manual annotation by the user. From these data points, the positions of the limbs were defined by the pixels in the relevant limb regions with the maximal distance to the body centre. When occlusion due to overlapping limbs, occurred a recovery step was used to distinguish between regions. Then the relative motion between successive frames was calculated by an optical flow method. The average error of this method was found to be 8.21cm.

The current state of the art method in this area comes from the work of Hesse et al. In their original paper in 2015 [1] Hesse proposed a system based on a random ferns body part classifier to estimate 21 joint positions. This was similar to that used by the Kinect API [28]. A synthetic infant body model was used to render a large data set of labelled RGB-D images. From this, a pixel-wise body part classifier was trained based on binary depth comparison of features. From the mean location of all pixels belonging to a body part estimated 3D joint positions were calculated. The system was evaluated on an adult data set [45] as well as against manually annotated 3D joint positions of a real infant sequence, they reported an AJPE of 13cm and 4.1cm respectively. However large errors were found on average for the left hand (14.9cm) and left shoulder (7.3cm). This was attributed to the synthetic training data set not including poses seen in the real test infant sequence.

This work was built upon [46] by the same group by including a feature selection step, generating more representative poses in the training data, integrating kinematic chain constraints and applying principal component analysis in torso pixels to correct for body rotations. These improvements led to a reduction in the average error of 0.6cm.

In one of their most recent papers [27], they proposed a model-based system. This is learnt from RGB-D data by optimising an objective function similar to that of Olsen et al. [41] but including additional terms such as integrating prior probabilities of plausible shapes and poses.

2.2.4 Evaluation Metrics

A further contribution of Hesse et al. to the field is the recent release of a publicly available benchmark data set of infant RGB-D videos called MINI-RGBD [7]. As can be seen in the preceding sections a wide range of methodologies have been proposed with a variety of evaluation metrics quoted when deployed on a range of different data sets. Therefore quantitative comparison between methods was difficult due to the lack of a publicly available benchmark dataset. However, the dataset is synthetic and so the potential for it to be unrepresentative of real infant movements and contain artefacts from the generation process should be considered.

Hesse et al. went on to compare the notable methods in infant pose estimation on their MINI-RGBD data-set and so provide benchmark metrics for the performance of the SOTA methods in this domain. This was done based on three metrics; average joint position error (AJPE), PCKh 1.0 and PCKh 2.0. PCKh is a commonly used metric in pose estimation problems [47] and is defined as the percentage of keypoints (joints) within a factor of the distance from the head to neck joint called the head segment length. In adults, PCKh 0.5 is a common metric but due to infants different anatomical ratios, such as their relatively short head to neck length, Hesse et al. suggested the use of PCKh 1.0 and PCKh 2.0.

From these comparisons, in the RGB 2D case the best performing system was found to be an adaption of the SOTA deep learning adult pose estimation OpenPose model [48, 2]. In which the model's output was offset by the average differences in an adult and infants anatomy. In the 3D case, Hesse et al.'s random fern methodology [46] was found to be the SOTA. This methodology is dependent on RGB-D video input.

2.3 Adult Pose Estimation Techniques

One of the most active areas of computer science research in recent years has been that of deep learning. This approach uses end to end learning to train models instead of domain specific handcrafted features. Furthermore, a great advancement in image classification and detection tasks was the use of convolution as an architectural prior [49]. This greatly reduced the number of learnable parameters as well as explicitly incorporating local spatial coherence. This has lead to an improvement in the SOTA in many problem domains [50], including adult pose estimation.

The DeepPose model by Toshev et al. [3] was one of the first to apply a convolutional neural network architecture to this problem space. Since 2014, when DeepPose showed SOTA performance, research in this domain has shifted from so-called 'classical methods' to exploring different deep learning architectures and approaches. This section reports the advancement of 2D and 3D pose estimation since this landmark paper.

2.3.1 2D Pose Estimation

As mentioned in the introduction to this section, DeepPose was one of the first CNN-models applied to this domain. The model architecture proposed used an AlexNet backbone [51] with an additional final layer which outputted $2k$ keypoint coordinates where k is the number of keypoints. The model was trained using an L2 loss function.

A novel approach of DeepPose was the refinement of its predictions using cascaded regressors. The model made an initial rough pose estimate. The input image was then re-cropped using these coarse keypoint estimates and the cropped images used as the inputs for the next stage of the model. These cropped images had higher resolution than the original leading to an increase in the precision of the model's predictions [3].

A major shift in the approach to 2D pose estimation since DeepPose [3] has been the use of heatmap targets instead of directly regressing keypoint coordinates. Tompson et al. [52] was the first to propose such a methodology. The model outputted a heatmap for each keypoint represent the probability of the keypoint being present at that pixel location, as shown in Fig 2.2. The final keypoint co-ordinates were found as the argmax of each heatmap. The targets for training were produced by placing a 2D Gaussian with $\sigma = 1.5$ pixels at the ground-truth (x,y) keypoint location. The mean squared error between the target and output heatmaps formed the loss function for training a CNN architecture. This heatmap approach was shown to outperform direct regression and so has been used by the majority models since. It is of note though that the use of the argmax function to transform heatmaps to final keypoint position is not differentiable. Therefore heatmap models are not fully end to end.

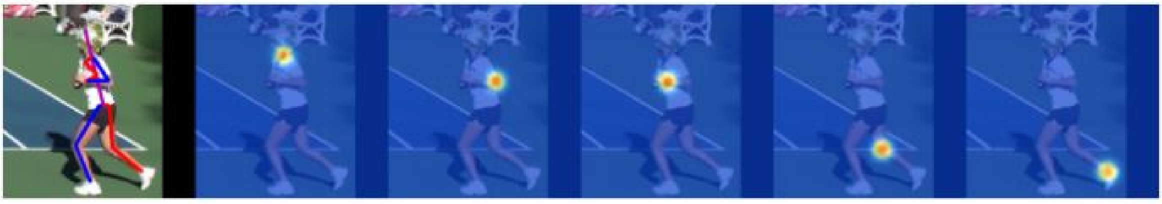

Figure 2.2: A visualisation of the heatmap methodology proposed by Tompson et al.[52].

Advancements in the 2D pose estimation domain, since the work of Tompson et al., have focused on the CNN architecture rather than the alternative approaches to the intrinsic problem. One such architecture is the stacked hourglass network [53]. So called as its design stacks several sub-networks consisting of pooling then upsampling layers which diagrammatically resemble an hourglass, see Fig 2.3. The motivation for the network design was the need to capture information on a variety of scales in order to accurately estimate a pose. For example, features such as hands require local evidence whilst large scale features such as a body's orientation require a global outlook. By varying the resolution of layers this variety in scale can be captured. The stacked hourglass architecture outperformed all previous models on a range of benchmark datasets.

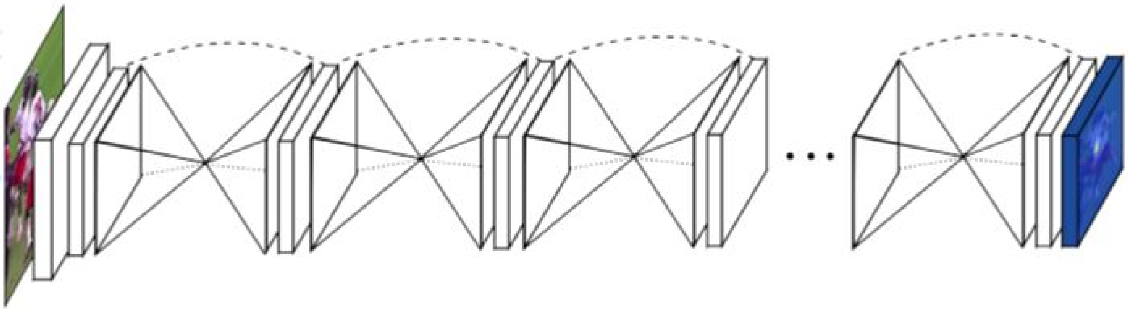

Figure 2.3: Summary of the stacked hourglass architecture [53].

However, the stacked hourglass approach required intermediate supervision, multiple skip connections and an intrinsically complex network design. In response, Xiao et al. [8] investigated the performance of a simple architecture based upon the same principles of the use of CNNs to predict keypoint heatmaps. A ResNet backbone [54] pre-trained on the Image-Net dataset [55] was connected via its final convolutional layer to a series of deconvolutional layers to upsample the low-level representations to heatmap resolution. Surprisingly, the performance of this network was found to be equal to if not surpass the performance of the more complex stacked hourglass model.

The current SOTA approach, HRNet (High-Resolution Network) [56], takes a different architectural approach to its predecessors. Instead of processing high to low to high-resolution representations of an input image, like in the stacked hourglass model and Xiao et al., HRNet maintains a high-resolution representation throughout the process. As the depth of the network increases a growing number of lower level resolution sub-networks are added in parallel, as shown in Fig 2.4. There is repeated exchange of information across these multi-resolution sub-networks. This approach was shown to outperform all models previously described in this section on several benchmarks.

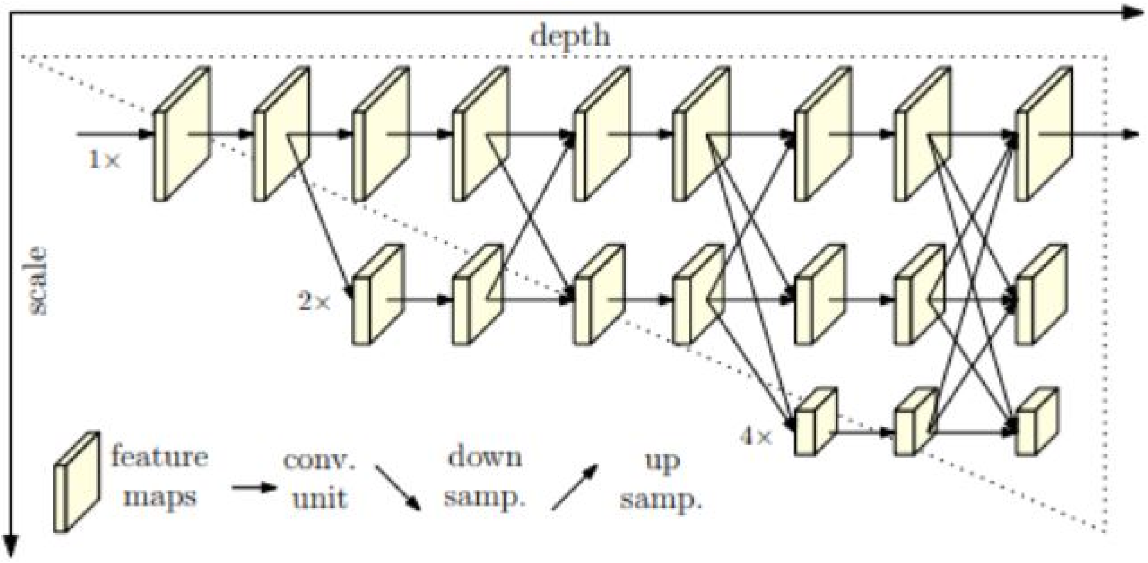

Figure 2.4: Summary of the parallel HRNet architecture [56].

In summary, since the emergence of the use of deep learning in the 2D pose estimation domain the use of heatmaps over directly regressing keypoints coordinates has become widely accepted as the currently optimal strategy. However, the precise CNN architecture to achieve this is still an area of active research. Several models of varying complexity and performance have been proposed such as the Stacked Hourglass, Xiao et al. and HRNet.

2.3.2 3D Pose Estimation

The most common approach to 3D pose estimation is to use a supervised approach. Although the generation of in the wild datasets is difficult as most data is generated indoors using a MOCAP system [5] these models have the highest level of performance. One of the earliest approaches outlining a supervised deep learning approach to the task was proposed by Li et al. [57] A simple network analogous to the work of DeepPose [3] in the 2D domain was proposed consisting of 9 convolutional and 3 fully connected layers. The model undertook two tasks; the detection of the presence of a keypoint and the regression of its position. The convolutional part of the model was trained using both task's loss function. This model outperformed all previous classical approaches.

This approach was built on by the work of Tome et al. [5]. This approach uses transfer learning to train a network derived from the Resnet-101 architecture [54], see Fig 2.5. Transfer learning applies features and representations learnt on a task with an abundance of available training data to a task for which data is scarce [58]. This is possible as it has been found unrelated tasks can share low and middle-level CNN features which are extracted in the early layers of a network [59].

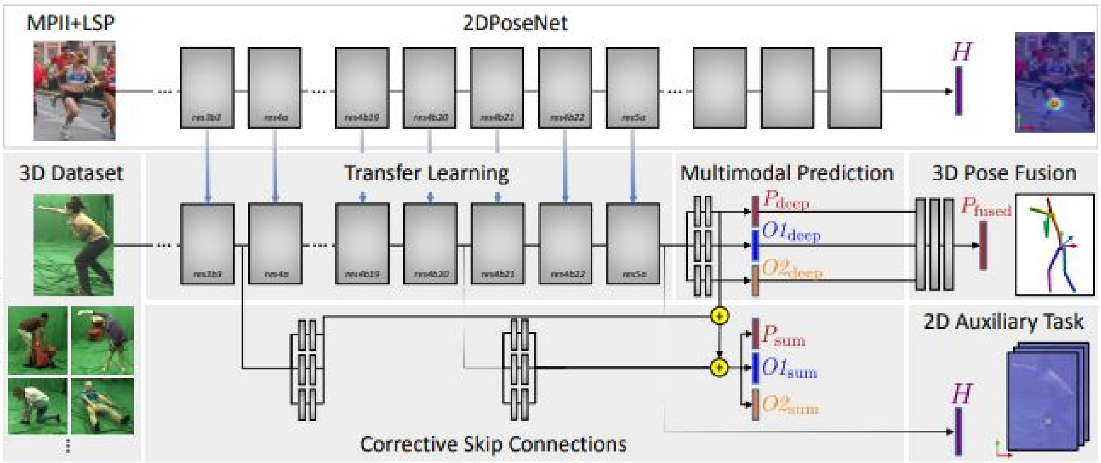

Figure 2.5: The model architecture used by Metha et al. ResNet-101 is used as a base network for transfer learning [5, 54]

Their training methodology firstly initialised the network's weights by training it for an image classification task on the large ImageNet dataset [60]. The learning rates of the early layers of the network were then reduced and a fully connected stub added to the end of the network. Further training was then conducted performed on pose estimation task using a 2D then 3D dataset. The relatively small size labelled 3D data-sets currently available means that the application of transfer learning in this domain resulted in SOTA results [5].

As mentioned earlier the majority of 3D pose estimation datasets are generated through the use of MOCAP and so models trained on such data do not generalise well to in the wild scenarios. However, large 2D in the wild datasets [6] do exist and hence models that generalise well to such scenarios have been developed. This inspired an alternative approach by Chen et al. [61]. They proposed splitting the model into 2 parts, as can be seen in Fig 2.6. The first a generic 2D pose estimation model using the CNNs described in Section 2.3.1. The second a non parametric nearest neighbour model that paired the estimated 2D pose to the closest 2D pose from a dictionary of paired 2D and 3D poses, formed from MOCAP data, hence predicting a 3D pose. This created a model that harnessed the available 3D ground-truth dataset but was still generalisable to in the wild scenarios.

Martinez et al.[62] took this a step further replacing the 3D dictionary lookup model with a deep learning 3D lifting network that took the estimated 2D pose as an input. As the lifting network took an array of keypoint coordinates instead of an image a simple and intuitive model was developed using deep learning features such as RELU, dropout and batch normalisation instead of a complex CNN. This lifting net-

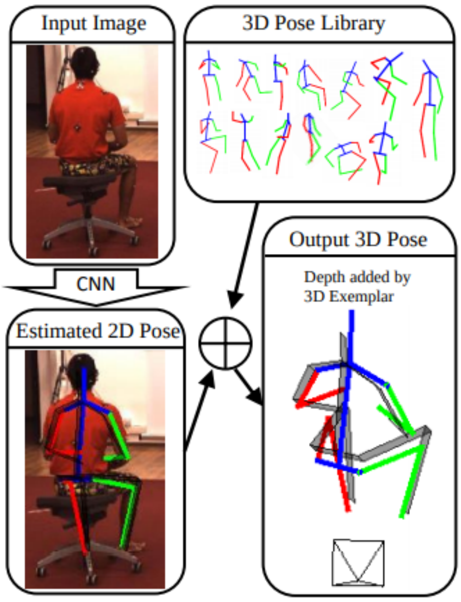

Figure 2.6: The 2 model methodology of 2D pose estimation followed by pose matching used by Chen et al. [61]

work was first trained on 2D ground-truths and then finetuned on the outputs of the SOTA 2D pose estimation model at the time of publication, the stacked hourglass model [53]. The initial motivation of the paper was to investigate whether the major source of error in 3D pose estimation stemmed from the 2D pose estimation or 3D lifting task. However, the authors were surprised to find this method surpassed all previously proposed end-to-end models.

Noticeable, the methods outlined so far have not incorporated the use of heatmaps as seen in the 2D domain, see Section 2.3.1. This is because the natural extension of such a method to 3D is the use of voxel representation of a 3D gaussian [63]. However, this dramatically increases the demands of the already memory-intensive process of heatmap generation. The current SOTA approach was one of the first approaches to successfully solve this problem. The major advancement MargiPose [64] was the use of 2D marginal heatmaps to capture the information of a voxel representation but with much lower memory requirements. Intuitively these 2D marginal heatmaps are the 2D projection along each axis of the voxel that would be generated, this can be seen in Fig .2.7. Another innovation was the use of the differentiable soft argmax function [65]. This made the end to end model fully differentiable and so an L2 loss function could be used directly regress the keypoint locations. As previously mentioned this method currently represents the SOTA in 3D adult pose estimation with leading performance on several benchmark datasets.

A small number of unsupervised methods have been also been proposed. For example, Rhodin et al. [66] proposed an unsupervised method which learns a geometry-aware body representation. This approach requires a set of synchronized multi-view images of the subject. The network then attempts to map one view to another and from this forms an encoding of the pose and scene geometry.

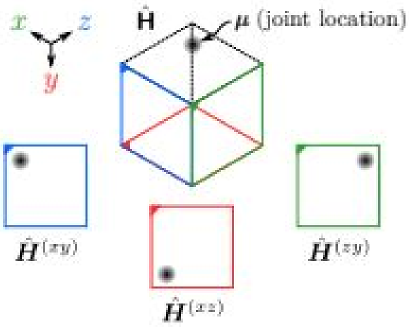

Figure 2.7: Shows the intuition of the use of marginal heatmaps by MargiPose. Each heatmap is a 2D view of a spherical 3D Gaussian which when put together can fully describe the original gaussian. [64]

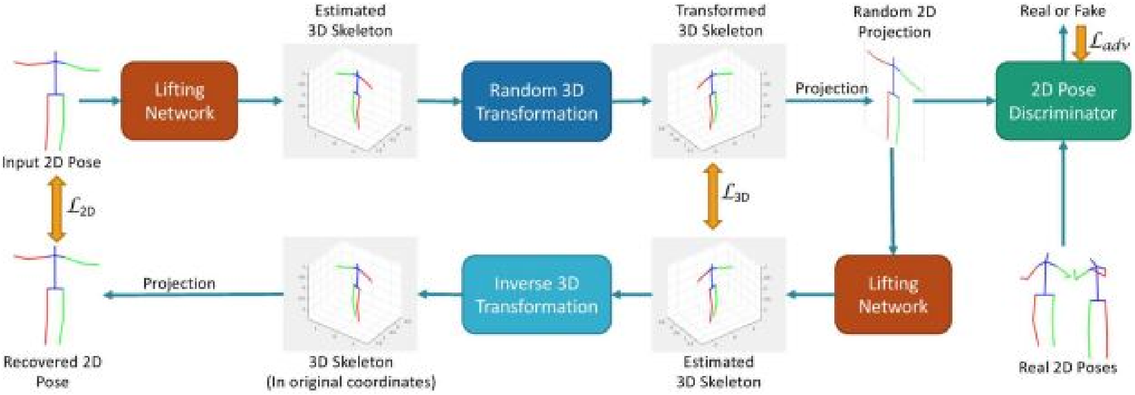

Figure 2.8: Chen et al. unsupervised 3D pose estimation pipeline with loss calculated according to geometric self-consistency, a 2D pose discriminator and temporal discriminator (not shown) [67]

The SOTA unsupervised method was proposed by Chen et al. [67]. This approach takes a 2D pose estimate and uses a lifting network to generate a 3D pose estimate. During training, this 3D skeleton is then re-projected by a random camera viewpoint generating a new synthetic 2D pose. This synthetic pose is then re-lifted to 3D and projected back to 2D using the original camera viewpoint. Given a perfect lifting network, this original and final 2D pose would be identical. This geometric self-consistency constraint is used to train the lifting network without supervision. This method was further improved through the addition of a temporal and a 2D pose discriminator to the loss function. These were trained using generative adversarial networks [68]. The full pipeline is outlined in Fig 2.8. Evaluation of the method showed better performance on benchmark data-sets than many leading weakly supervised methods. The addition of supervised fine-tuning led to comparable results to the SOTA supervised approach of the time [67, 5].

2.4 MAVEHA

The previous section outlined the recent success of deep learning techniques in the area of adult pose estimation. It also described the current mainly classical or non-specialist models developed for the infant domain. The Imperial College London based MAVEHA project, therefore, sees an opportunity to radically improve the infant SOTA by the adaptation of adult pose estimation deep learning models. This section outlines some of the recent work conducted by the group.

2.4.1 Data Acquisition

A data collection stage has been conducted. Using the Azure Kinect DK [28] an RGB-D dataset of 25 babies has been recorded. The babies were also fitted with electromagnetic sensors to ascertain the 3D ground-truths. However, as outlined in Section 2.2.1 wearable sensors are prone to calibration and technical issues. In this case, an inhomogeneous magnetic field introduced complex errors into the sensor data. Efforts are currently underway to rectify this corrupted data but are as yet unsuccessful. Given the current global pandemic, further data acquisition within the period of this project was not feasible. It is of note that relative to the large amount of training data required for deep learning architectures [50] the MAVEHA data set is small.

2.4.2 2D Transfer Learning Results

MAVEHA's first, unpublished, example of the application of deep learning to the infant pose estimation domain is a 2D estimation model which uses a transfer learning-based approach [5] to adapt the simple baseline model proposed by Xiao et al. [8] and described in Section 2.3.1.

The model was pre-trained on the MPII Human Pose [6] data-set before final training on the MAVEHA infant data-set, for which 2D ground-truths exist. A Resnet50-based faster RCNN model [54] was also used to extract an initial bounding box around the infant. The final evaluation of this approach is still underway but the initial results are promising.

2.5 Legal and Ethical Considerations

The main legal and ethical considerations for this project arise from the use of multiple human image datasets to train and evaluate pose estimation models. As will be outlined in Chapter 3 this was done using the following datasets; MPII [6], MPI_INF_3DHP [5], MINI-RGBD [7] and the MAVHEA real infant dataset. The first three datasets are published online, freely available for academic use and the collection of the data was performed with the relevant ethical considerations.

The MAVHEA dataset has been collected by the team working at Imperial College

London who have experience in medical imaging research and adhere to clinical governance standards. To collect this dataset MAVEHA used consent forms and acquired additional approval from Imperial's and King's College London's ethics committees. All videos taken are fully anonymous and the use of such anonymous patient image data has been classified as ethically uncritical by the NHS National Institute for Health Research (NIHR) and follows ICO standards. The experiments undertaken in this project, including viewing images, evaluating numeric data, and judging them with qualitative image analysis, are non-invasive and safe. Furthermore, the supervisors of this project are trained in Good Clinical Practice and the handling of patient data.

2.6 Summary

This chapter has outlined the need for an accurate pose estimation model as part of the wider research effort to automate the general movement assessment of infants.

A review of the literature revealed that the current approaches in infant pose estimation are based on classical methods such as random ferns [1]. This is a distinct contrast to the adult domain in which the SOTA approaches since DeepPose in 2014 [3] have used modern deep learning techniques.

Therefore this work aims to extend scientific literature by investigating if deep learning techniques developed in the 3D adult pose estimation domain could be successfully adapted to the problem of 3D infant pose estimation via transfer learning. This will be done by developing a baseline 3D infant pose estimation model which can then be tested on the MINI-RGBD dataset against the current SOTA approach.

Chapter 3

Methodology

The methodology of this project has been developed on the basis of the findings of Chapter 2. The natural extension to MAVEHA's 2D infant pose estimation model based on the work of Xiao [8] would be to add a 3D lifting network inspired by the work Martinez et al.[62] However, due to the limited size of infant datasets a transfer learning approach would be required with pre-training of both models on the larger adult datasets.

To achieve this three distinct machine learning models have been developed namely; an infant bounding box model, a 2D adult and infant pose estimation model and a 3D adult and infant lifting network. All infant models have been trained on the MINI-RGBD dataset [7] whilst the 2D and 3D models use the MPII [6] and MPI_INF_3DHP [5] datasets respectively. This process is outlined in Fig 3.1 and explained fully in this chapter.

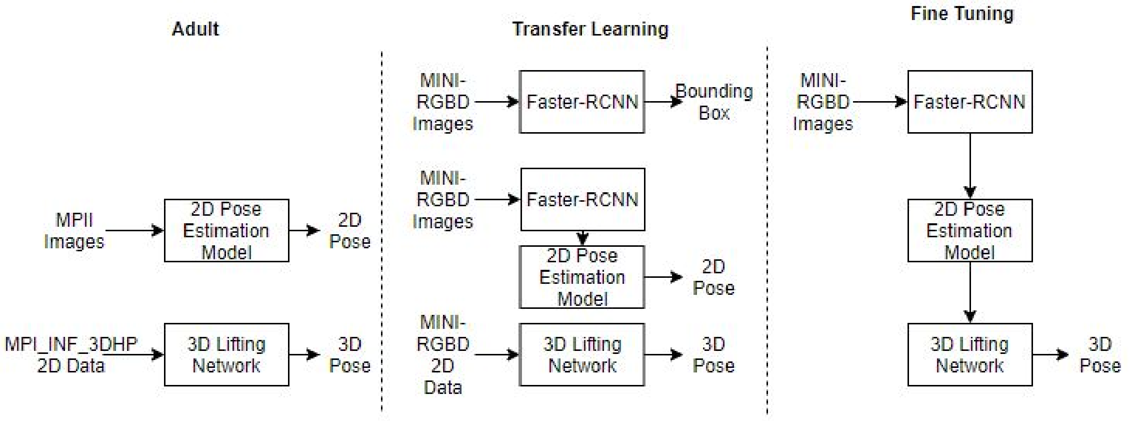

Figure 3.1: Shows the three stages of the training pipeline of the 2D pose estimation, 3D lifting network and bounding box models. Not shown is that the 2D pose estimation and Faster-RCNN models were pre-trained on the ImageNet dataset.

Formally, for a given RGB image input, I, the 3D pose estimation model outputs a series of k keypoint coordinates in 3D space $\mathbf{y} \in \mathbb{R}^{3k}$. The aim is to learn a function f^* that for a dataset of N poses minimizes the prediction error, $\mathcal{L}$:

$$f* = \min_{f} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(I) - \mathbf{y}_i). \tag{3.1}$$

As described, f^* is a two-stage deep neural network, made up of a 2D pose estimation network, g^*, and a 3D lifting network, z^* such that,

$$f^*(I) = (z^* \circ g^*)(I) \tag{3.2}$$

These networks are similarly defined as,

$$g^* = \min_{g} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(g(I) - \mathbf{x}_i), \tag{3.3}$$

$$z^* = \min_{z} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(z(\mathbf{x}_i) - \mathbf{y}_i), \tag{3.4}$$

where **x** is a series of k keypoint coordinates in 2D space $\mathbf{x} \in \mathbb{R}^{2k}$.

3.1 Bounding Box Model

An early design consideration when building a CNN, such as the 2D pose estimation model, is the relationship between dimensions of the input and output of the model. This is because the dimensions of the output from a convolutional layer is not only determined by the properties of the layer such as its kernel size, stride and pooling but also the size of the input. Therefore to produce an output of a consistently valid dimensionality the input image size must be fixed [51].

A naive approach would be to resize each image using processes such as nearest neighbour or bi-linear interpolation [69]. However, a more effective technique is to instead crop the picture around the object of interest to the task, in this case an infant, and then scale this section of the image to the required size using the aforementioned techniques. This not only gives an image input of the required fixed size but also reduces the variation in the scale of the object of interest as well as removing extraneous information from the scene. For these reasons the use of cropping as a preprocessing step for image inputs into 2D pose estimation models has been ubiquitous since Toshev et al.'s [3] use of such a technique in their original landmark paper.

To achieve this a model is required to output the top left and bottom right corners, $\{(x_1, y_1), (x_2, y_2)\}$, of the box with the minimum area which bounds the object of interest entirely. This box is parallel to the x and y axis to reduce the degrees of freedom to 4. This equates to finding the maximum and minimum pixel locations along each axis associated with the object of interest.

For the MPII dataset this could be simply calculated by finding the maximum and minimum 2D ground-truth keypoint locations positions along the x and y axis then adding an arbitrary pixel padding of 100 pixels. Using the ground-truth is allowed in this case as the aim is to maximise the learning of the 2D pose estimation model which is then transferred to the infant model. Comparison against SOTA models or the deployment of the model in the adult domain is not a consideration.

However, these conditions do not hold when designing the 2D infant pose estimation model to make inferences on the MINI-RGBD dataset. Therefore a SOTA deep learning model has been trained to perform the task.

3.1.1 Model Architecture

Research into object detection models has developed two main families of models; YOLO [70] and R-CNN [71]. In general R-CNN models are found to be slower but have higher accuracy than YOLO models. As real-time performance is not a consideration for this model as all videos are pre-recorded and the primary aim is to optimise the model for accuracy, an R-CNN model is used.

A recent iteration of the R-CNN family is the Faster R-CNN architecture [71] which builds upon the previously published Fast R-CNN model [72] shown in Fig 3.2A. The Fast R-CNN model takes an input image along with a set of regions of interest (ROI), generated by a different model. These RoI are passed through a pre-trained CNN to extract low level features. The output of the CNN is then passed to an RoI pooling layer. Then finally through a series of fully connected layers, the model outputs a class prediction via a softmax layer and a bounding box via a linear output. This is repeated for each region of interest.

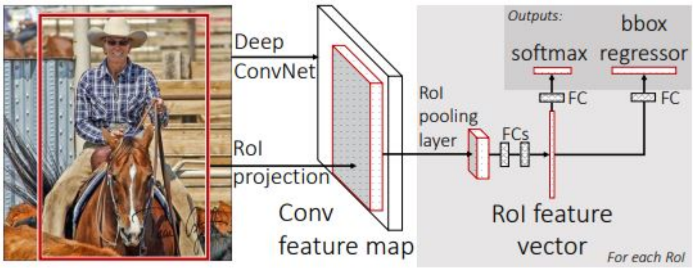

Figure 3.2: A summary of (A) the Fast-RCNN architecture [72] and (B) the Faster-RCNN architecture [71]

As can be seen in Fig 3.2B the Faster R-CNN model couples a region proposal network (RPN) to the Fast R-CNN model to create a single model design. The output of the deep CNN, the same as is used by the Fast R-CNN, is passed to a small network which outputs multiple RoIs and a binary class prediction of the presence of an object within that region. These RoI are then used as the input to the Fast-RCNN model along with the deep CNN features.

The Faster R-CNN architecture shows SOTA performance on multiple benchmarks as well as winning multiple competitions. This high level of performance along with the success of the MAHVEA project using the same architecture on their real infant dataset is the reason that the Faster R-CNN architecture has been selected.

3.1.2 Training

The ground-truths on which to train the Faster R-CNN model are generated from the segmentation mask of the infant supplied in the dataset. Due to the synthetic nature of the dataset these masks have no noise. Therefore the ground-truths can be found by finding the first and last non zero indices when the mask pixel values are summed horizontally and vertically. This gives the max and min values of the infant's location in both the x and y direction from which the bounding box can be formed as $(x_{min}, y_{min}), (x_{max}, y_{max})$. Padding is then added to these indices.

To train a Faster R-CNN model on the MINI-RGBD dataset an initial pre-processing step is required. The ground-truth bounding box is labelled '1' as only object type, an infant, being identified. The other class '0' is applied by the model when no object was present in the proposed RoI. The training dataset is augmented by the random application of a flip and rotation operation using an affine transformation [73].

A Faster R-CNN model pre-trained on the ImageNet classification task [55] is then download from the PyTorch model zoo [74] and the final layer is modified to predict only two classes. Due to the relative simplicity of the task pre-training on an adult dataset is not required.

A training schedule is then set up with MINI-RGBD videos 1 to 8 the training set, 9 to 10 the validation set and 10 to 11 the test set. An Adam optimiser [75] with a learning rate of 1^{-4} is used, with a batch size of 4. The loss function for the training set gives a cumulative loss based upon the three tasks that are undertaken by the Faster-RCNN model, as described in Section 3.1.1. Therefore the more intuitive intersection over union (IoU) metric [76] is used to evaluate the performance of the model on the validation set. To avoid over-fitting, if for three successive epochs there is no improvement in the validation set's IoU training is ceased.

3.2 2D Pose Estimation Model

The design of the 2D pose estimation is based closely on the work by Xiao et al. [8] as the MAHVEA project has previously shown that such an approach can be successfully adapted to the infant domain. Fig 3.3 outlines visually the workflow described in this section.

3.2.1 Pre-Processing

The initial task is the pre-processing of the input image. The input image is first cropped in accordance to the bounding box generated either from the 2D keypoint ground-truths or the Faster R-CNN model depending on the dataset as outlined in Section 3.1. From the centre and scale of this bounding box, the image is cropped and scaled by an affine transformation [73] to give an output image of 256x256 pixels. This size has been chosen in line with Xiao et al.'s [8] original work and also to simplify the later up-sampling process. If the image is part of the training set a random rotation is added to the transformation to augment the data.

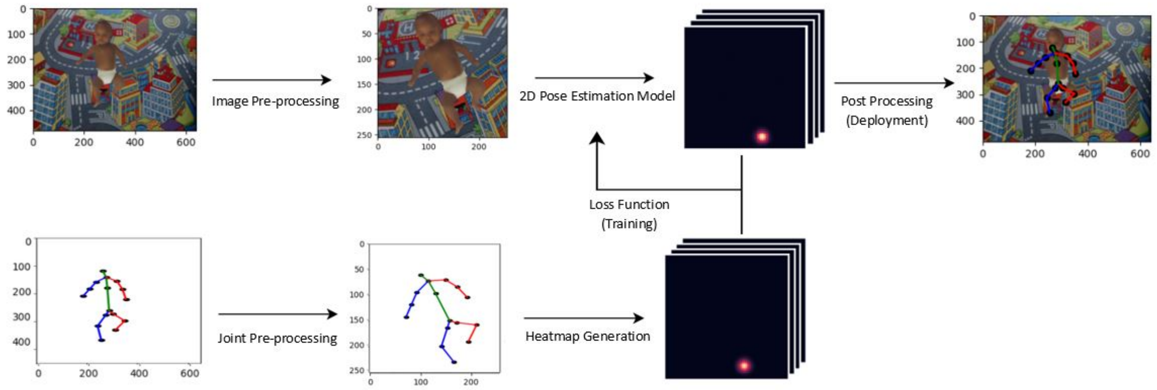

Figure 3.3: Data pipeline for the training and deployment of the 2D pose estimation model based upon the work of Xiao et al.[8]

Next, if the model is being trained the 2D keypoint ground-truths also undergo pre-processing. Firstly the identical cropping and rotation affine transformation is applied to each 2D ground-truth. Next non-visible keypoints, defined as those with ground-truths outside the frame of the image, are calculated. These keypoints are not included in the later calculation of losses during training or in the evaluation metrics.

As outlined in the Section 2.3.1 the model architecture chosen for the 2D pose estimation task, Xiao et al. [8], outputs a heatmap representing the probability of a keypoint being present at each pixel rather than directly regressing the keypoint's 2D position. Therefore the final pre-processing step is to generate the ground-truth heatmaps. For each keypoint, a $\{64 \times 64\}$ heatmap is produced by placing a 2D gaussian with a σ value of 2 at the keypoint's scaled 2D ground-truth location.

3.2.2 Model Architecture

The architecture used mirrored that developed by Xiao et al. [8] This network uses ResNet [54], a common network used for image feature extraction, as a backbone. This network aimed to provide a solution observed degradation of deep learning models as more layers were added, shown in Fig 3.4A. This is caused by the possibility that the repeated multiplication of the gradient during backpropagation could result in the gradient in early layers becoming infinitely small. ResNet's solution is the use of residual units, such as the one shown in Fig 3.4B. These units have so-called ”identity shortcut connections”. Therefore the performance of a deeper model should be no worse than that of a shallower model as it can be formed by stacking layers of identity mappings onto it's shallower counterpart.

The depth of the ResNet backbone is, therefore, a trade-off between increasing training time and accuracy. The results of Xiao et al. [8], see Table 3.1, showed a modest increase with networks deeper than ResNet50. It was decided that this loss in accuracy would be outweighed by the gain in iterative design time during the project.

To the last convolutional layer of the ResNet50 architecture 4 deconvolutional layers

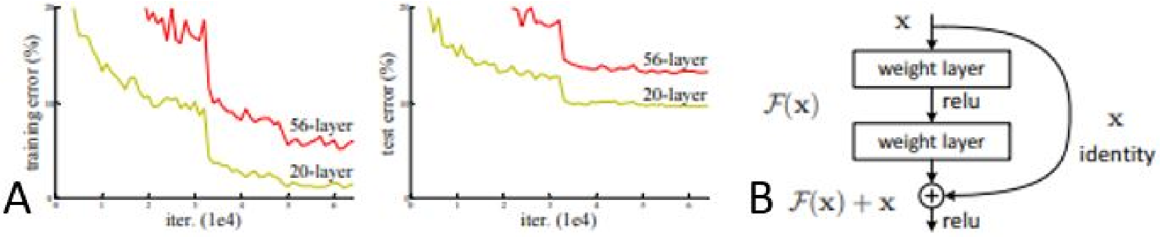

Figure 3.4: A) Graphs taken from the original ResNet paper showing the degradation in model performance with increasing model depth. B) Shows ResNet's solution in the form of a residual learning building block. [54]

Architecture	MPII PCKh/%
ResNet50	88.532
ResNet101	89.131
ResNet152	89.620

Table 3.1: Results from Xiao et al.[8] shows the small increase in accuracy with increasing ResNet backbone depth

were added, as can be seen in Fig 3.5. The first three layers each had 256 filters, a 4x4 kernel size, a stride of 2 and a padding of 1. The final layer had k number of filters, a kernel size of 1x1 and a stride of 1 where k is the number of predicted keypoint positions. This resulted in an output of k $\{64 \times 64\}$ pixel heatmaps representing the probability that the keypoint was present at that pixel location.

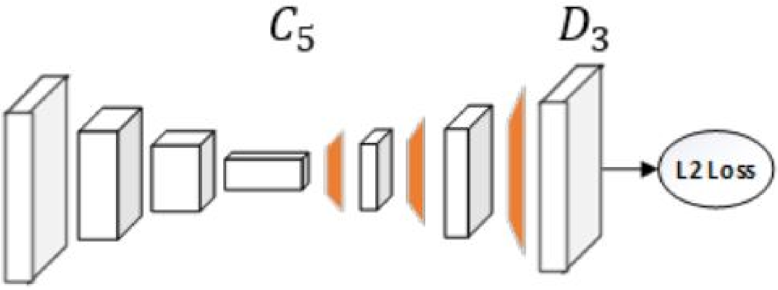

Figure 3.5: The Xiao et al. 2D pose estimation architecture used in this project. The final 1x1 deconvolutional layer is not shown. [8]

3.2.3 Post-Processing

If being deployed the keypoint locations are then found from the heatmap in a post-processing step. The key point positions in the pre-processed 256x256 image are found as the max prediction in each of the k heatmaps. The initial pre-processing transformation is then inverted to give the 2D keypoint locations on the original image as a $\{k \times 2\}$ array.

3.2.4 Training

To train the 2D pose estimation model firstly a Resnet50 model pre-trained on the ImageNet classification task [55] is download from the PyTorch model zoo [74] and the deconvolutional layers described in Section 3.2.2 are added.

The model is then trained on the MPII dataset [6] followed by the MINI-RGBD dataset [7]. The same training, validation and test set split are used for the MINI-RGBD dataset as in Section 3.1.2. For the MPII dataset, this split is predefined. Again training is exited if there was no improvement in the validation set loss for three successive epochs. An Adam optimiser [75] with a learning rate of 1^{-4} is used, with a batch size of 16. The loss between the ground-truth and model output is found by the Mean Squared Error (MSE) loss [77] with non-visible keypoints, as defined in Section 3.2.1, not included in the calculation.

3.3 3D Lifting Network

The 3D lifting network design is based on the work of Martinez et al. [62]. This work aimed to separate the 2D and 3D lifting network to identify which of the processes was the major source of error in the system. This could not be achieved using a more mainstream end-to-end system. The surprising result from this work was that coupling this lifting network with a SOTA 2D pose estimation model led to a simple, scalable and efficient model which rivalled complex end-to-end SOTA deep learning models.

3.3.1 Pre-Processing

The input to the network is a list of flattened 2D keypoint coordinates of length $2k$. In line with the standard protocol in 3D pose estimation, the 3D ground-truth poses are zero-centred around the pelvis joint [4]. Data augmentation via rotation and translation of the 3D ground-truth and relative correction to the input 2D pose was not implemented and so could be an area for future improvement.

3.3.2 Model Architecture

The model architecture proposed by Martinez et al.[62] is simple but based upon recent developments in deep neural networks.

Firstly a linear layer is applied to the input to increase its dimensionality to 1024. Next two residual units, shown in Fig 3.6, are applied. Each of which are made up of two Rectified Linear Units (RELUs) [78] and not convolutional layers like many 3D Pose Estimation models. This due to the low-dimensional nature of 2D keypoint input data compared to the usual image input data. Therefore less computationally expensive RELUs can be used to add non-linearities to the network. Inspired by the ResNet architecture [54] a residual connection spans the 2 linear units, Martinez et al. [62] found this to reduce the error rate by 10%. A final linear layer then produces an output of size $3k$.

However, Martinez et al.[62] found that when this architecture was trained on 2D ground-truths and then tested on the output from a 2D pose estimation model the generalisation was poor. Therefore further features were added to the residual units to stabilise the training process. Firstly the maximum norm of the weights of each layer were constrained to be less or equal to 1. Secondly, dropout [79] of 50% was used during training. Finally, batch normalisation [80] was used. This achieved the desired stabilisation of training at the cost of a small increase in training time.

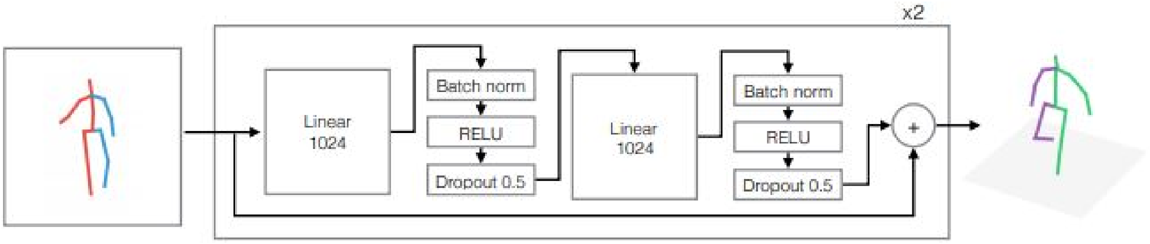

Figure 3.6: Martinez et al.'s 3D lifting network architecture. Not shown are the initial and final linear layers. [62]

3.3.3 Training

The lifting network is trained firstly on the adult MPI_INF_3DHP dataset [5] followed by the 2D ground-truths from infant MINI-RGBD [7]dataset. An additional training step of fine-tuning the model on the outputs of the 2D infant pose estimation model of Section 3.2 completes the training process.

A similar training protocol to those described in Section 3.1.2 and Section 3.2.4 is used, with identical MINI-RGBD dataset splits and training exit criteria.

An Adam optimiser [75] with an exponential learning rate scheduler [81] is used, in line with the methodology of Martinez et al.[62] The respective learning rate and γ hyper-parameters are given in Table 3.2. These are found by a brief hyper-parameters search enabled by the fast training time of the lightweight 3D lifting network. The loss between the ground-truth and model output is found by the Mean Squared Error (MSE) [77].

Input Dataset	**Learning Rate**	γ
MPI_INF_3DHP	10^{-3}	0.96
MINI-RGBD ground-truth	10^{-2}	0.9
2D Pose Estimation Model Output	10^{-2}	0.9

Table 3.2: Shows the adam optimiser learning rate and exponential learning rate scheduler γ hyper-parameters values used to train the 3D lifting network with various input datasets

3.4 Transfer Learning

The extension from the work of Xiao et al. [8] and Martinez et al. [62] is the use of transfer learning to perform pose estimation in the infant domain. The standard approach to this technique is to train a model in a related domain with a large image dataset size then change the input domain to the required smaller dataset and retrain the model. This results in a final model with increased accuracy as the initial weights when training on the final smaller dataset are closer to the optimum values than if random initialised when not pre-trained. This method can require the replacement of the final layers of the model if the output dimensions or task in the new domain is also shifted [82].

In this project transfer learning is achieved by pre-training pose estimation models on the adult MPII and MPI_INF_3DHP datasets and then fine-tuning the models on the infant MINI-RGBD dataset. The size of each of these datasets is summarised in Table 3.3.

Dataset	Number of samples
MPII	54,087
MPI_INF_3DHP	131,592
MINI-RGBD	10,000

Table 3.3: Summary of the size of the datasets used in this project

However, this section outlines why for several reasons, a standard transfer learning approach fails when shifting the domain of the 2D pose estimation model and 3D lifting network from adult to infant. It then goes on to outline the mapping solution developed to solve this problem.

3.4.1 3D Lifting Network Problems

Firstly, consider the 3D lifting network outlined in Section 3.3. For this network, the input is not an image like the majority of computer vision deep learning models but a vector of length of $2k$. Therefore as the adult MPI_INF_3DHP dataset and infant MINI-RGBD dataset have a different number of labelled keypoints the size of the input as well as the output to the network changes with the domain shift. This problem could be solved by replacing both the initial and final linear layers of the network with new randomly initialised linear layers of the correct size.

However, the change in the input structure is not only in the length of the vector but also the ordering of keypoints. For example, the left ankle is at position 0 in the MPI_INF_3DHP dataset but position 8 in the MINI-RGBD dataset. This is critical as the lifting the network has no concept that the input and output of this network are 2D and 3D poses. Instead, it sees the inputs and outputs as multidimensional vectors for which it aims to find a functional relationship between. Therefore this seemingly trivial swap leads to an entirely new weight mapping being required by the network to perform its task. This severely limits the efficiency of the transfer learning process as the network has to relearn associations between input and output dimensions instead of fine-tuning the proportions of the body which is the main difference between the domains in this case.

3.4.2 2D Pose Estimation Problems

A problem also arises in the transfer learning of the 2D pose estimation model. Here the input is fixed in size due to the pre-processing steps outlined in Section 3.2.1. However, the output size was dependent on the number of labelled keypoints. The initial standard approach as outlined in the introduction to this section was to, therefore, replace the final deconvolutional layer network to output the correct number of heatmaps.

However, the adult MPII dataset that is used for pre-training contains fewer labelled keypoints than the MINI-RGBD dataset. Therefore in the pre-training on the adult dataset feature representations for keypoints, such as fingers, that are only labelled in the infant dataset do not minimise the loss function and so are not learnt. When training is then transferred to the infant domain the losses are dominated by the failure of the model to identify the extra keypoints. Therefore, the changes to the model weights via backpropagation attempt almost exclusively to find representations for these extra keypoints. However, as the infant dataset is small this has limited success. Hence the need for transfer learning in the first place. Moreover, the fine-tuning of the recognition of keypoints present in both domains is limited due to large losses of the missing keypoint.

3.4.3 Mapping Solution

A simple solution is proposed to solve the problems outlined in the previous two subsections. Each dataset's keypoint frameworks is mapped to one common definition, d, of length k. This removes the variability in number, presence and order of the keypoints that results in the problems in transfer learning previously outlined.

The mapping of a dataset, x of length n, is defined by a list, Q, which contains the corresponding index of each keypoint of d in x. Q is formed to minimise the distance between d and the mapped dataset,

$$Q = \overset{q}{\min} \sum_{i=0}^{k} \mathcal{L}(x_i - y_{qi}) \text{ where } q_i \neq q_j. \tag{3.5}$$

Here $\mathcal{L}$ represents a qualitative evaluation of the difference between two keypoint labels made through the assessment of sample visualisations and the keypoint names. This Q, is then used to form a linear map, f, from $\mathbb{R}^n \to \mathbb{R}^k$.

$$f(x) = Mx, \tag{3.6}$$

where M is defined as,

$$M_{ij} = \begin{cases} 1 & \text{if } j = Q_i \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

The mathematics presented here formalises a simple and intuitive mechanism. For example, if the left ankle keypoint is at index 1 in d and index 8 in x, $Q_1 = 8$.

The constraint applied in Eqn 3.5 ensures a one-to-one mapping. This is required as the pose estimation models used in this project are designed to output one dimension per keypoint. For this condition to hold $n \leq k$. The implication of this is that the training dataset with the fewest keypoints must be chosen as the common definition. In the case of this project that is the MPII dataset with 16 keypoints[6]. A visualisation of the output of the mapping process can be seen in Fig 3.7.

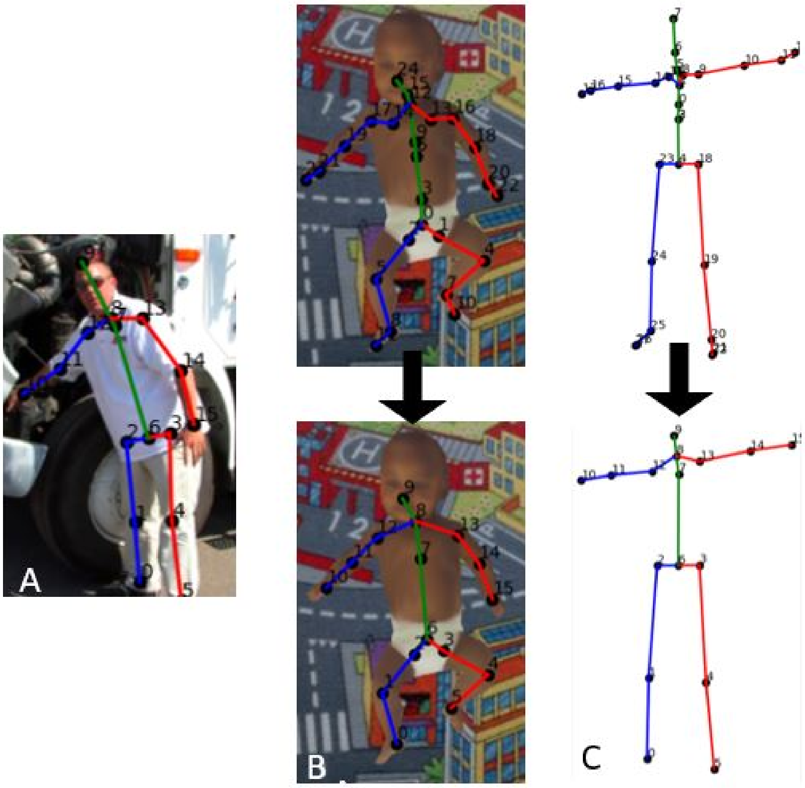

Figure 3.7: Shows visually the mapping of the MINI-RGBD, (B), and MPI_INF_3DHP, (C) keypoints to the MPII keypoints to provide a unified framework. (A) shows a sample from the MPII dataset for comparison.

3.5 Summary

To summarise, this section has outlined the training of a Faster-RCNN infant bounding box model, 2D infant pose estimation model inspired by Xiao et al.[8] and 3D infant lifting network based on the work of Martinez et al. [62]. It has also explored the challenges faced in transfer learning between the adult and infant domain and presented a solution by using one unified keypoint framework between all datasets used in the training process.

Chapter 4

Implementation

The methodology outlined in Chapter 3 has been implemented in Python 3.6. This was accomplished using the machine learning framework Pytorch [83] along with other standard data and image processing frameworks including NumPy [84], Matplotlib [85] and OpenCV [86]. This chapter outlines the major implementation sections of this project namely; Datasets, Models, Training, Inference and Evaluation. If the reader wishes to reproduce or build upon this implementation the open-source Github repo for this project can be found at `https://github.com/simonEllershaw/3DInfantPoseEstimation`

4.1 Datasets

The use of transfer learning in this project's methodology means that samples from multiple different datasets are required to be loaded. Moreover, 3 distinct machine learning models have been trained. Therefore, depending on which model is being trained or used for inference, differing inputs and outputs types can be required from the same dataset source.

An efficient implementation of the loading of each dataset has been achieved by extending Pytorch's Dataset class [87], as can be seen in Fig 4.1. This abstract class requires the implementation of the '__getitem__' and '__len__' methods. This allows the object to be passed to Pytorch's DataLoader class [88] which has the ability to load multiple sample batches in parallel.

In this project, Pytorch's Dataset class is extended by the JointsDataset class. The decision was made to eagerly load as much of the dataset as possible due to the large amount of generic pre-processing that has to occur, for example the calculation of bounding boxes. To achieve this the JointsDataset defines an empty Python array as the database property of the class and an abstract '__get_db' method. The return value of '__len__' is set to the length of this database array.

The JointsDataset class is then extended to encompass each of the four Dataset types that are required by the project; BboxDataset, Joints2DDataset, Joints3DDataset and

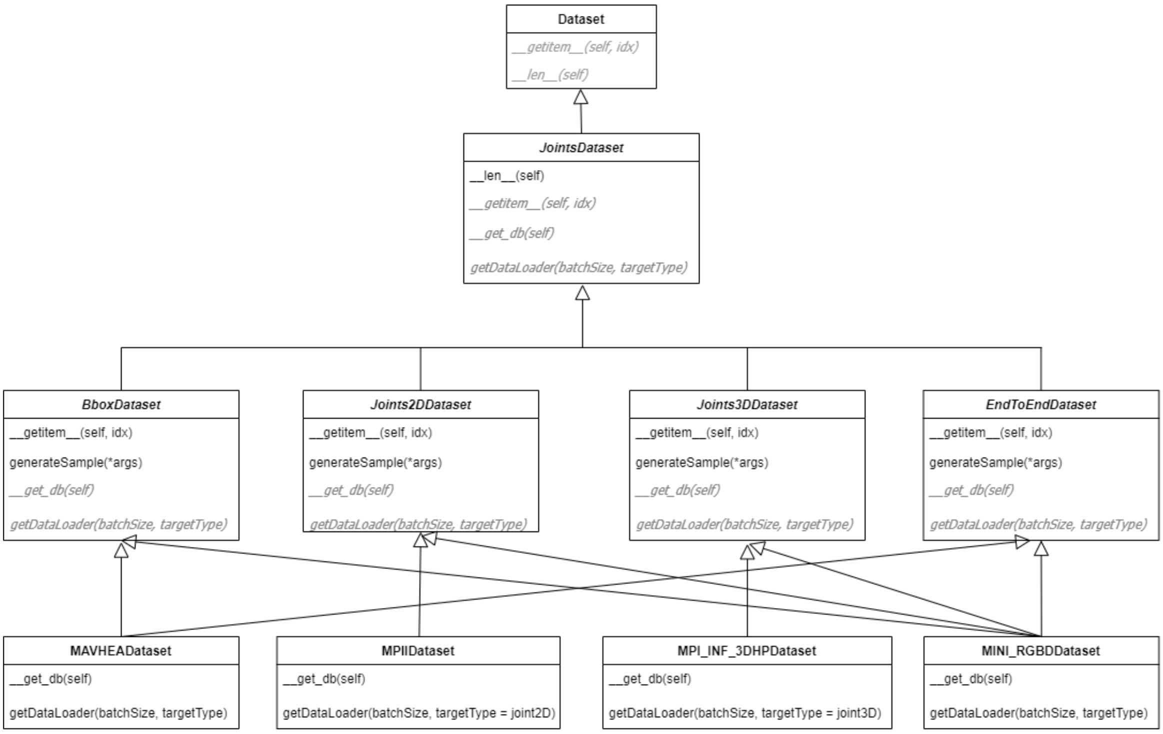

Figure 4.1: A UML diagram showing the inheritance structure of the dataset implementation extending the parent Pytorch Dataset class [87]. Abstract functions are shown in grey italics. Helper functions and class properties are not shown

EndToEndDataSet. The input and output data types for each of these Datasets is summarised in Table 4.1. At this level of the inheritance hierarchy two key methods are implemented. Firstly, 'generateSample' defines the data required to append a sample to the Dataset's database. For example, the 2D joint positions, path to the image file, PCKhThreshold along with the centre and scale of the subject (derived from the bounding box) are required for each sample added to the Joints2DDataset database. The other key method implemented at this level of the hierarchy is the '__getitem__' method. This is achieved by indexing the object's database and then conducting the final pre-processing stages, which cannot be done eagerly. These include the loading of RGB images into memory and the random augmentation of data. The sample is then returned to the caller, in most cases a DataLoader.

Dataset	**Input**	**Output**
BboxDataset	Image	Bounding Box
Joints2DDataset	Image	2D Keypoints
Joints3DDataset	2D Keypoint	3D Keypoint
EndToEndDataSet	Image	3D Keypoint

Table 4.1: Summary of the input and output data for each child of JointsDataset

The leaves of the hierarchy tree are the dataset specific classes. These implement the final two abstract methods. Firstly, the '__get_db' fetches and appends all samples from the specific dataset into the class' database list in accordance to its parent's 'generateSample' method. Pre-processing steps specific to that dataset such as the generation of bounding boxes or the mapping of joints to MPII format are performed here. Crucially this method also defines the training, validation and testing split for the dataset. Finally, 'getDataLoader' instantiates a dataset of each data split and embeds them within a dataLoader which is then returned.

Although the inheritance hierarchy, outlined here and visualised in Fig 4.1, could be further refined to form a more elegant solution it has a number of beneficial properties and behaviours. Firstly, due to the 'generateSample' method, the data structure for a sample of a given Dataset type is independent of the source of the data. This allows the interchange of datasets with no update to the model or training code. Secondly, a base dataset can inherit from multiple parent classes. This makes the interchange of data input and output types a matter of a change of constructor argument. For example, the MAVHEADataset implementation inherits from both the EndToEndDataset and the Joints2DDataSet. Thirdly, the use of the Pytorch Dataset [87] to instantiate DataLoader [88] classes provides multi-threaded loading of data coupled with automatic batching. These behaviours greatly decrease the data loading time as well as the training and inference time of models. Finally, the framework set out here is easily extensible as all pre-processing and loading procedures are abstracted away from dataset specific implementation. Therefore the addition of a new dataset requires the implementation of just two methods, one of which 'getDataLoader' is trivial.

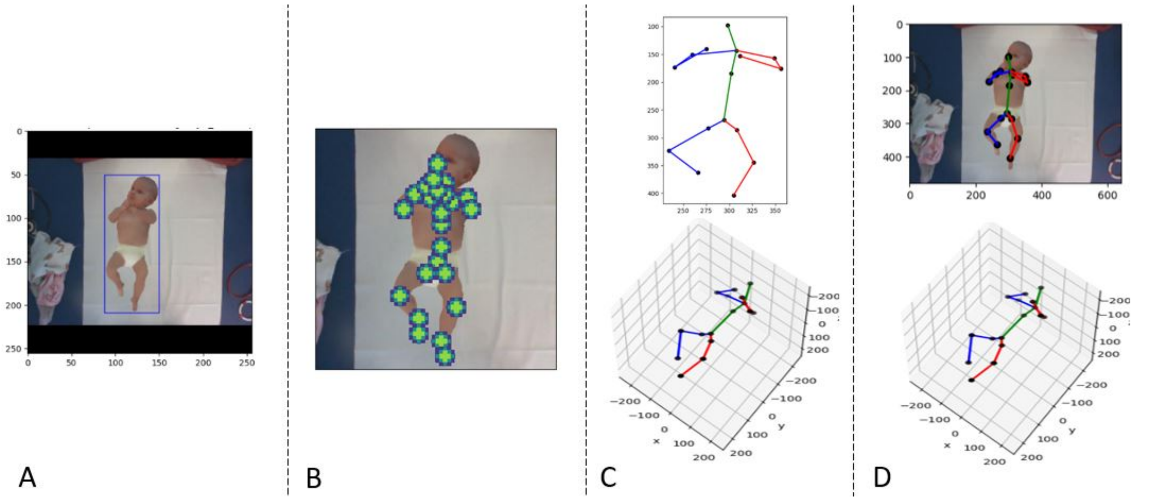

Figure 4.2: An visualisation of the same sample from the MINI-RGBD dataset [7] with differing parent datasets. A) BBoxDataset, B) Joints2DDataset, C) Joints3DDataset D)EndToEndDataset.

Not shown in the Fig . 4.1 are the large number of helper functions contained within each class. These include methods for various pre-processing steps including the generation of 2D joint heatmaps, the mapping of joints to MPII format and calculating a bounding box from 2D joint positions. Notably, each Dataset has a method to visualise a sample using Matplotlib [85]. This provides useful debug insights as well as the basis for the visualisations of the outputs from the trained models presented in Chapter 5. Fig 4.2 shows these visualisations for a single sample from the MINI-RGBD dataset [7]. It can be seen that image shows both the dataset's input and output.

4.2 Models

Three separate model architectures are used by the methodology outlined in Chapter 3. The implementation of these models in Pytorch are freely available for research purposes either from the codebase of the original papers [8, 62] or Pytorch's model zoo [74]. This section outlines the modifications that are required to adapt these models to give the functionality required for this project.

A Faster R-CNN bounding model pre-trained on the ImageNet dataset [55] is downloaded directly from Pytorch's model zoo [74]. The final layer is then replaced with a layer to predict only two classes; infant and no object. A complication of this model's implementation is that it expects the targets of each batch to be a dictionary of lists instead of the default list of dictionaries provided by Pytorch's DataLoader class [88]. Therefore, the following custom collate function is required for the bounding box DataLoaders,

```
def collate(batch):
        return tuple(zip(*batch))
```

The 2D pose estimation model, adapted from the work of Xiao et al. [8], is based upon a ResNet50 backbone. This is downloaded from the Pytorch model zoo [74] pre-trained on the ImageNet dataset [55]. The final adaptive average pool and linear layer are removed from the CNN. To this backbone the deconvolutional layers, described in Section 3.2.2 are added. These are implemented using Pytorch's 'ConvTranspose2d' layer as follows,

```
class Upsampling(nn.Module):
    def __init__(self, numJoints):
        super(Upsampling, self).__init__()
        self.convTrans1 = nn.ConvTranspose2d(2048, 256, 4, 2, padding=1)
        self.convTrans2 = nn.ConvTranspose2d(256, 256, 4, 2, padding=1)
        self.convTrans3 = nn.ConvTranspose2d(256, 256, 4, 2, padding=1)
        self.convTrans4 = nn.ConvTranspose2d(256, numJoints, 1, 1)

    def forward(self, x):
        x = F.relu(self.convTrans1(x))
        x = F.relu(self.convTrans2(x))
        x = F.relu(self.convTrans3(x))
        x = self.convTrans4(x)
        return x
```

Finally, the 3D lifting network implementation is taken directly from the codebase accompanying the work of Martinez et al. [62]. The only modification required is to the size of the inputs and outputs of the model. All other hyper-parameters, including the size of the linear layers, number of residual units and percentage dropout, are not changed from those found to be optimal by Martinez et al. [62]. It is of note that no pre-trained model was downloaded for the 3D lifting network.

4.3 Training

All the models in this project are trained using the mechanism of back-propagation to minimise a loss function. This occurs in epochs which use the training set for back-propagation and the validation set to avoid over-fitting. Each of these datasets are then split into batches and loaded onto a GPU to decrease computation time. At the end of each epoch if the total loss of the validation set represents a 1% or greater improvement over the previous best the current model weights are saved. However, if the loss has not improved for three consecutive epochs training is excited. At the end of each epoch the training and validation loss are logged as well as the current best loss and the time taken to process the epoch. The following skeleton code listing outlines this process,

```
def train_model(model, dataloaders, device, criterion, optimizer):
    while(True):
        best_loss = Inf
        for phase in [”train”, ”val”]:
            if phase == ”train”:
                model.train()
            else:
                model.eval()
            running_loss = 0.0

            # Training
            for source, targets, meta in dataloaders[phase]:
                source = source.to(device)
                targets = targets.to(device)
                outputs = model(source)
                loss = criterion(outputs, targets)
                if phase == ”train”:
                    optimizer.zero_grad()
                    loss.backward()
                    optimizer.step()
                running_loss += loss.item()

            # Epoch stats
            epoch_loss = running_loss / len(dataloaders[phase])
            if phase == ”train”:
                trainLoss = epoch_loss
            elif phase == ”val”:
```

```
            valLoss = epoch_loss
            # Improvement has to be at least by 0.1%
            if epoch_loss < best_loss * 0.999:
                best_loss = epoch_loss
                saveCheckpoint(model)

    logMetrics(epoch, trainLoss, valLoss, best_loss, epochTime)

    # Early exit if no improvement for 2 epochs on val set
    if numberEpochsWtNoImprovement > 2:
        break
```

Such a common training infrastructure could only be developed as all the datasets in the project are the extension of the same base class, Pytorch's Dataset [87], and the model architectures are implemented using Pytorch's machine learning framework. However, there are a number of differences between the training of the models arising from their separate methodologies including the loss function used and the presence of a learning rate scheduler. The differences are outlined in detail in Chapter 3. It is of note though that a specific implementation of Mean Square Error loss [77] is required to account for the non-visible joints in the 2D pose estimation datasets.

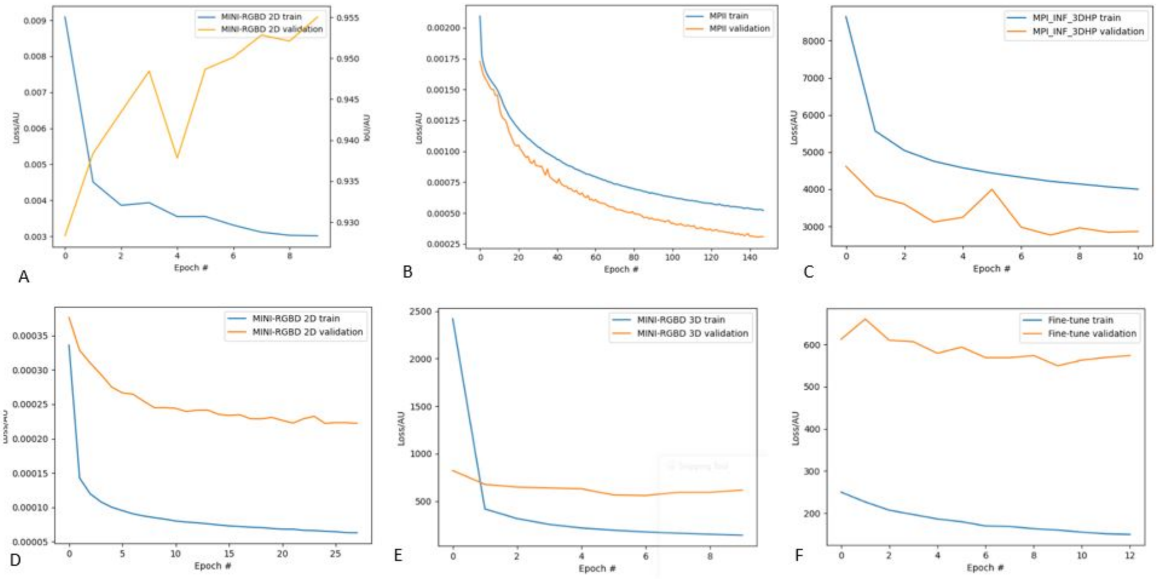

Figure 4.3: Plots of the training and validation loss of the A) Infant Bounding Box Model, B) MPII 2D Pose Estimation Model, C) MPI_INF_3DHP 3D Lifting Network D) MINI-RGBD 2D Pose Esimation Model, E) MINI-RGBD 3D Lifting Network F) MINI-RGBD End to End Finetuning. Note that in A) IoU loss was used to evaluate the performance of the model, this increases as the network becomes more accurate unlike loss functions.

All training was conducted on HP EliteDesk 800 G3 TWR with an Intel Core i7-7700K 4.20GHz processor and a GeForce GTX TITAN X graphics card. The metrics recorded for the training process are presented as training loss graphs in Fig 4.3 and Table 4.2 shows the average training time per epoch for each model.

Model	Average Training Time per Epoch/s
Infant Bounding Box Model	1176.20
MPII 2D Pose Estimation Model	805.64
MINI-RGBD 2D Pose Estimation Model	138.84
MPI_INF_3DHP 3D Lifting Network	80.41
MINI-RGBD 3D Lifting Network	2.19
MINI-RGBD End to End Finetuning	59.79

Table 4.2: Shows the average training time per epoch for each model.

4.4 Inference and Evaluation

Inference from the outputs of the bounding box and 3D lifting model requires a trivial reshaping of the output tensors and inverting any pre-processing steps. However, for the 2D pose estimation model it is more involved as the model predicts 2D heatmaps for each keypoint not their exact positions. Therefore a number of post-processing steps are required to find the arg-max of each heatmap and translate this to a pixel co-ordinate on the orginal image. The implementation for this process was taken from the codebase of Xiao et al. [8].

The evaluation metrics for the pose estimation models are calculated by iterating through the batches of inferences and ground truths for the test set of each dataset. The euclidean distance is then found for each keypoint, the AJPE. Furthermore, by thresholding the euclidean distances with respect to a PCKh threshold contained within the metadata of the sample the PCKh per keypoint is also found.

4.5 Summary

This chapter has outlined the specific implementation details of the methodology outlined in Chapter 3. This has included the dataset inheritance hierarchy which provides an extensible data loading framework and the methods used to adapt publicly available deep learning architectures to give the functionality required by this project. The training process for these models has also been outlined along with a presentation of the metrics resulting from this process. Finally an outline of the inference and evaluation process was given.

Chapter 5

Results

This section presents a set of quantitative metrics, including PCKh and AJPE defined in Section 2.2.4, as well a sample of outputs from the models described in the methodology.

5.1 Bounding Box Model

A Faster R-CNN model was successfully trained to perform object detection on the infant MINI-dataset. The average IoU for the test set is 87.7%. Shown in Fig 5.1 are a series of outputs from the model compared to the ground-truth including failure cases. It is of note that the failure case shown is rare with 0.0045% of test set images having an IoU value of less than 70%.

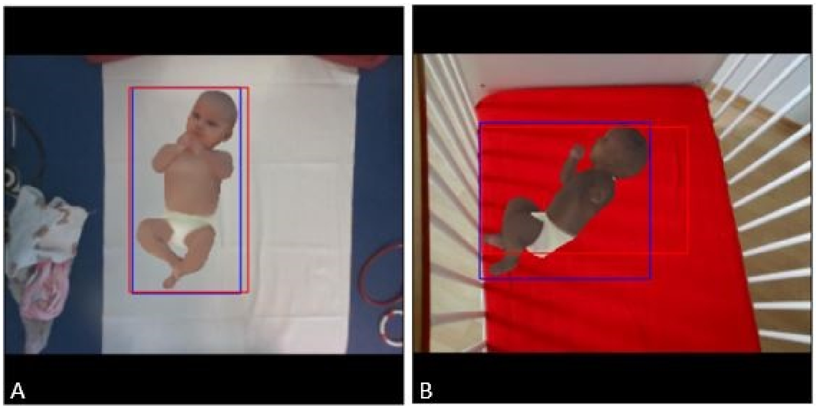

Figure 5.1: Visualisation of the bounding boxes outputted by the trained Faster-RCNN model. The blue box denotes the ground-truth bounding box and the red box the Faster R-CNN model's output A) Shows a typical successful output with an IoU value of 90.1%. B) and C) show to failure cases with IoUs of 67.0% and 0.0% respectively.

5.2 2D Pose Estimation Model

5.2.1 MPII Dataset

As outlined in Section 3.2.4 the 2D pose estimation model is firstly trained on the MPII dataset. No test set is held out so the validation set was used to gain intuition on the model's performance. On this dataset, an AJPE of 8.17 pixels is recorded and a mean PCKh of 98.87%. The PCkh threshold is taken as the distance between the neck and head keypoint multiplied by a factor of 0.5. It is also of note that the bounding box used to crop the image in the pre-processing step is generated using 2D keypoint ground-truths as outlined in Section 3.1.2.

A sample of model outputs from increasingly challenging inputs are shown in Fig 5.2 allowing for observational assessment of the performance of the model.

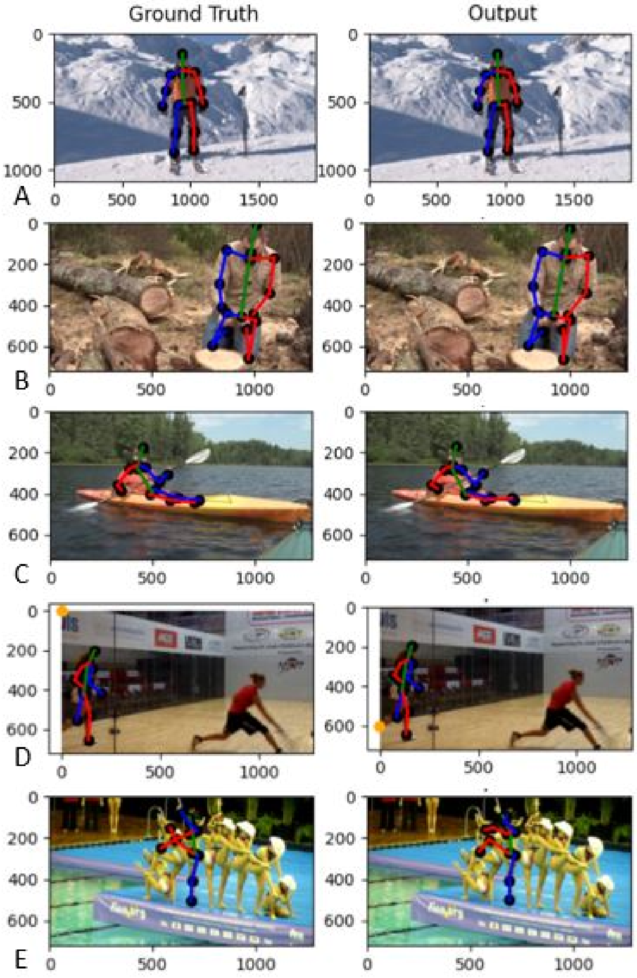

Figure 5.2: A selection of outputs from the 2D pose estimation model trained on the MPII dataset. All images are a member of the validation set. Yellow points represent keypoints that are outside the image range so according to the MPII dataset are 'non-visible'

5.2.2 MINI-RGBD Dataset

The MPII trained 2D adult pose estimation model is then trained on these synthetic infant MINI-RGBD dataset. On this dataset the AJPE is found to be 8.17 pixels and the mean PCKh is 93.77% and 96.05% at threshold factors of 1 and 2 respectively. The distance between the neck and head keypoint is given as a constant for each video in the dataset.

Hesse et al. [7] showed that the current SOTA 2D model for this dataset is an adapted version of the OpenPose pose estimation model[2]. The average PCKh of the OpenPose model for video 11 and 12, the test set for this project, are quoted to be 86.54% and 91.64% at thresholds factors of 1 and 2 respectively.

Fig 5.3 shows three sample outputs in comparison to the 2D ground-truth pose. Similarly to Fig 5.2 the inputs show an increase in complexity until the breaking point of the model is reached. The increasing complexity in this dataset is mainly due to self-occlusion from limbs as other factors such as multiple subjects and object occlusion are not present.

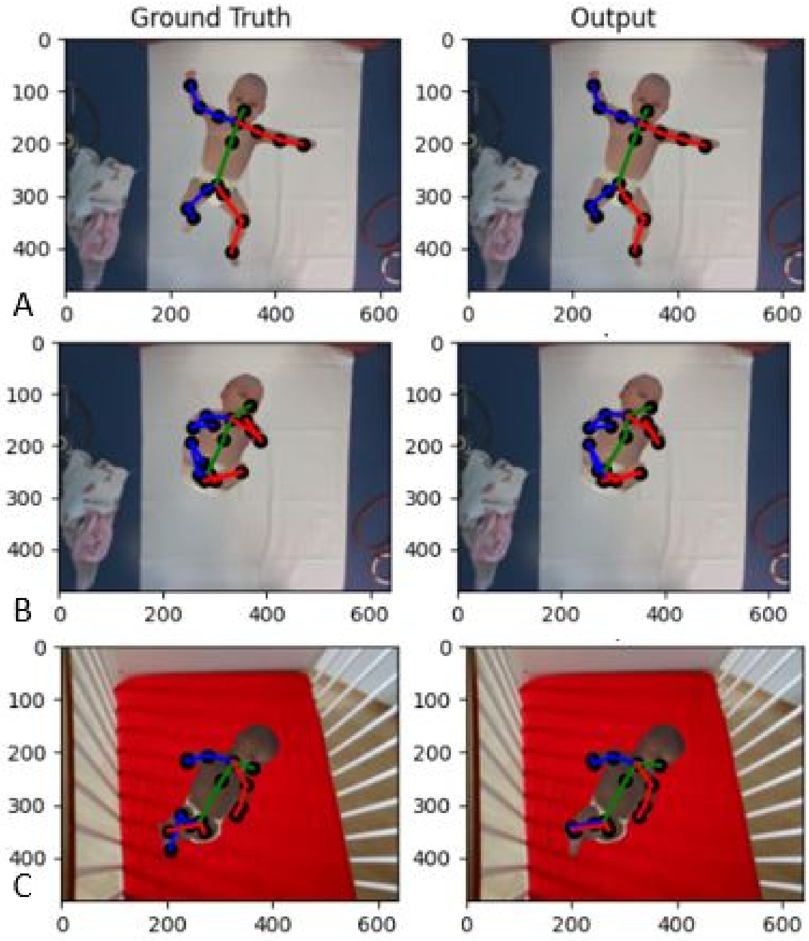

Figure 5.3: A selection of outputs from the 2D pose estimation model trained on the MINI-RGBD dataset. All images are a member of the test set.

Fig 5.4 shows a comparison of the OpenPose model to our model on a per keypoint basis. It is of note that comparison here is not like for like as Hesse et al. [7] used all videos, not just videos 11 and 12, to produce this data. This figure shows the superior performance of our model compared to the current SOTA for nearly every keypoint. Also shown is the variation of keypoint accuracy for our model.

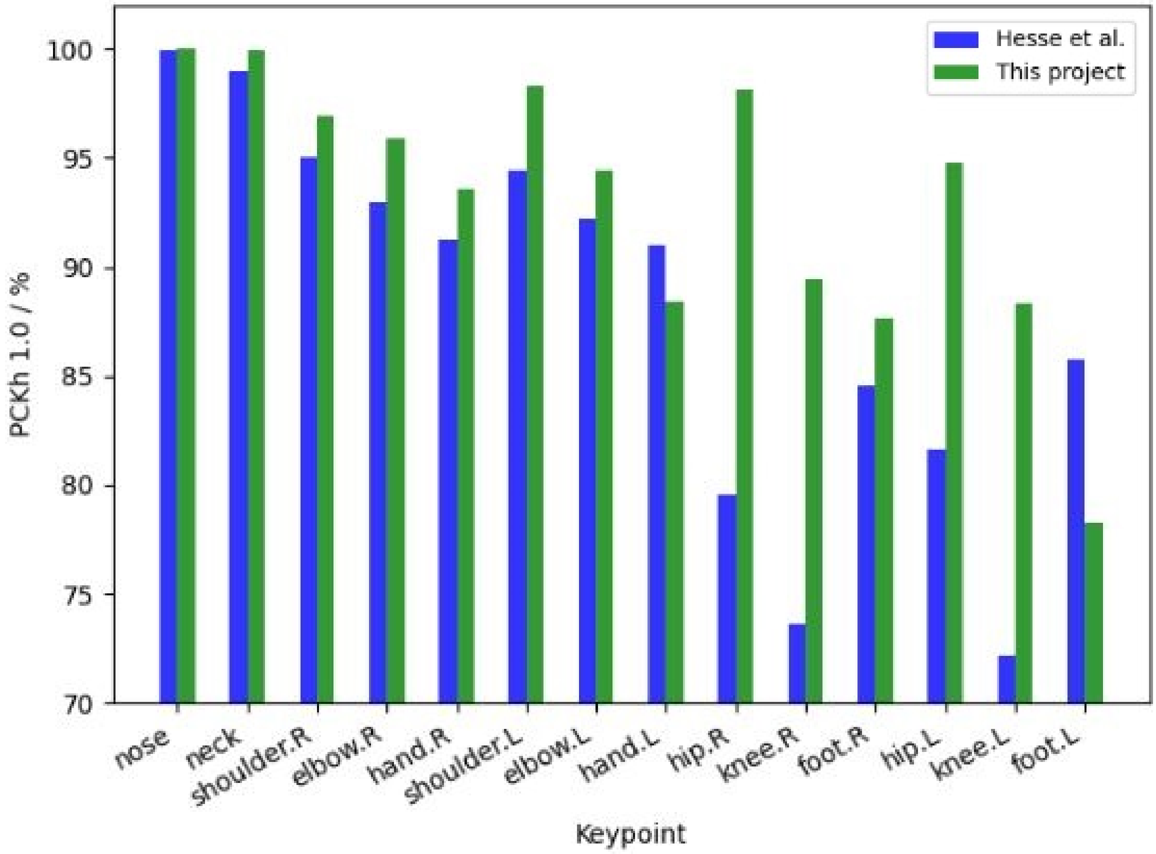

Figure 5.4: A bar chart comparing the PCKh 1.0 performance of the current SOTA adapted 2D OpenPose model, blue, and our 2D infant pose estimation model, green, on a per keypoint basis.

5.3 3D Lifting Network

5.3.1 MPI_INF_3DH Dataset

Pre-training of the 3D lifting network was carried out using the MPI_INF_3DH dataset. The resulting network has an AJPE of 64.02mm and a PCKh of 92.21% on the validation set. The PCKh threshold used was a constant 150mm as mandated by [5].

As can be seen in Fig 5.5 high PCKhs, greater than 98%, are recorded for central keypoints such as the pelvis, hip, neck, head and shoulders. As the 3D co-ordinates are in centred upon the pelvis co-ordinate these central keypoints have limited relative movement so to infer their relative position is a simple task of determining the orientation of the body. The outer limbs have greater freedom though and so are more challenging and hence the reduction in PCKh for these keypoints.

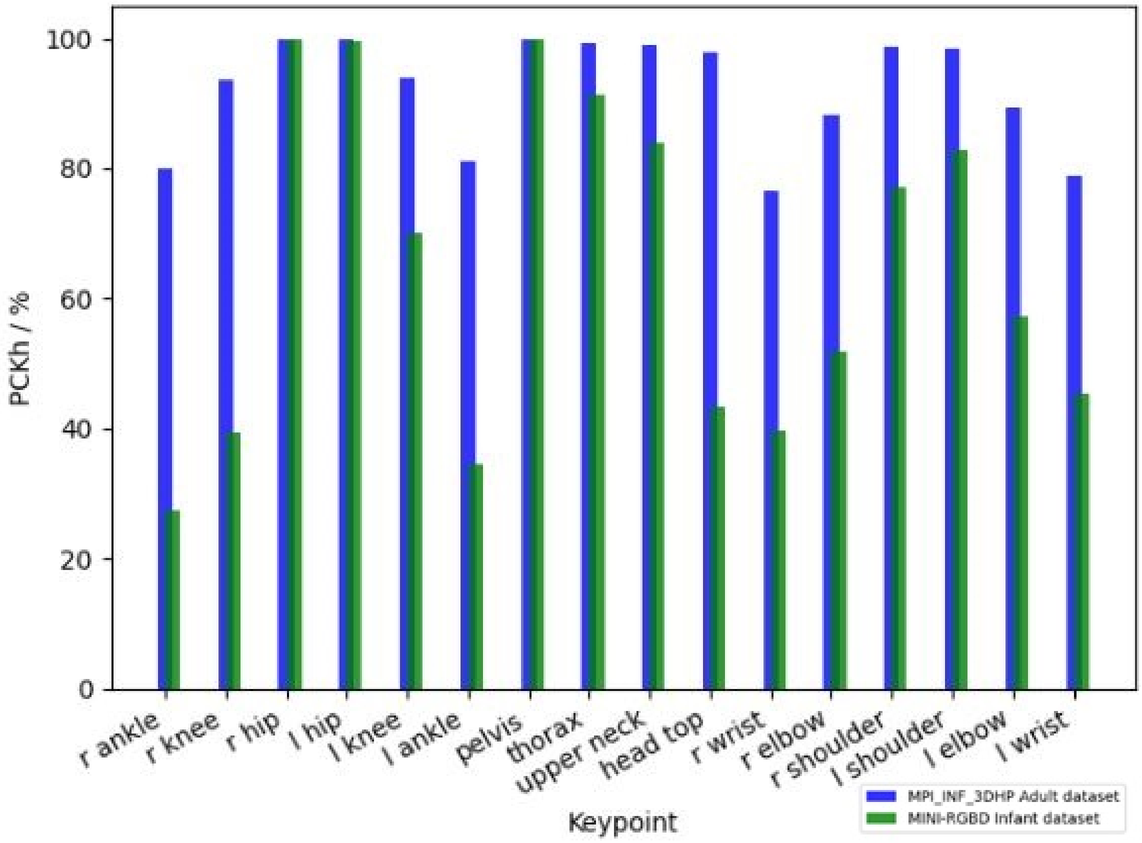

Figure 5.5: A bar chart showing the PCKh per keypoint for the 3D lifting model when trained on the MPI_INF_3DHP dataset, blue, and MINI-RGBD dataset, green.

5.3.2 MINI-RGBD Dataset

The final stage of the training process is to train the 3D pose estimation network on the MINI-RGBD dataset via transfer learning. This was done in two stages. Firstly the network is trained using the 2D ground-truths from the dataset and then finetuned using the outputs of the 2D pose estimation model, from Section 5.2.2, as the inputs. Sample outputs from this 3D lifting network are shown in Fig 5.6. The outputs of the model for the two test data videos have also been converted to video format and can be viewed at `https://www.youtube.com/watch?v=rvivVQzxUIc&feature=youtu.be`

The AJPE and PCKh metrics for these models are shown in Table 5.1. Also shown in Table 5.1 is the superior performance of this model compared to the metrics published for the same dataset using the current SOTA 3D infant method of random ferns [1]. It is also of note that the SOTA model results quoted here require an RGB-D input whereas the model developed in this project does not require a depth input.

As expected the 3D lifting network performs best on when trained and tested on the

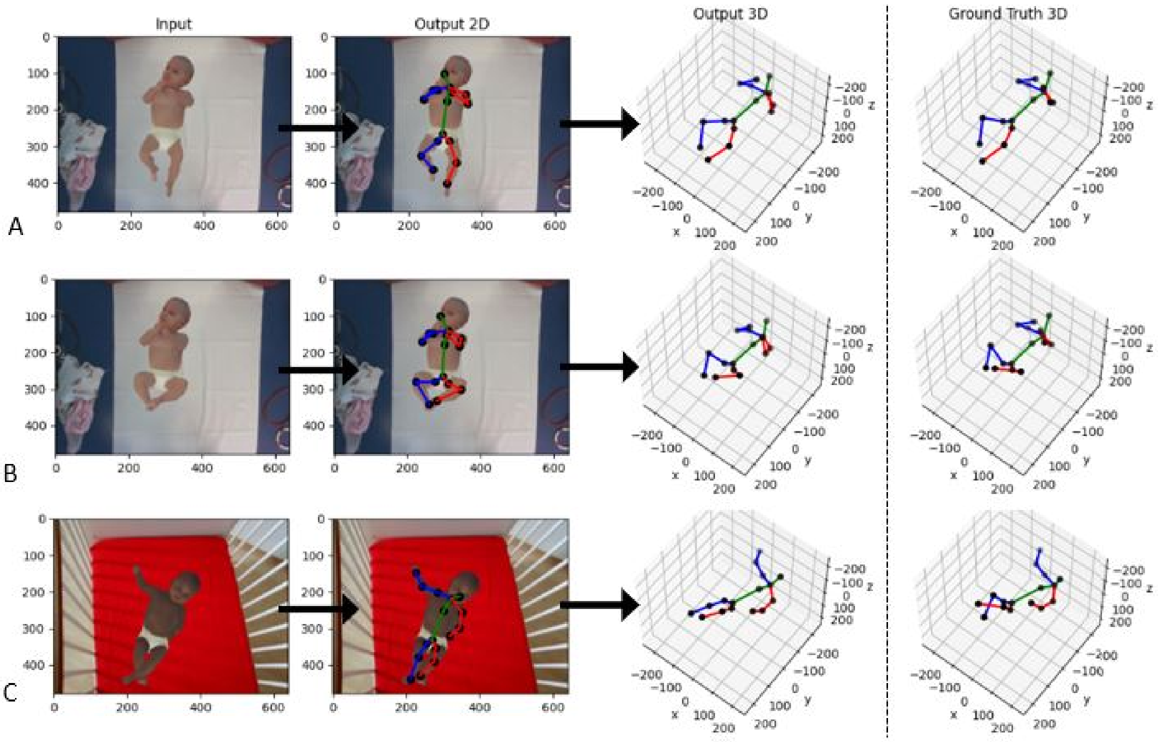

Figure 5.6: Figure showing a sample of outputs when the 3D pose estimation model is applied to images from the MINI-RGBD test set. [7]

Input Data	AJPE/mm	PCKh 1.0/%	PCKh 2.0/%
MINI-RGBD 2D ground-truth	25.44	69.28	91.19
2D Pose Estimation Model (No fine-tuning)	28.80	65.68	88.26
2D Pose Estimation Model (fine-tuning)	28.47	65.20	88.99
Hesse et al.	44.90	51 .09	83.87

Table 5.1: Shows the Average keypoint Error and PCKh for the 3D lifting network trained and tested on various inputs. Comparison metrics are shown for Hesse et al.'s SOTA random ferns model when tested on the same videos[7].

MINI-RGBD 2D ground-truths. A small degradation in performance is seen when the 2D pose estimation model is used as input data. fine-tuning, by training on these inputs, counters this degradation to a limited extent.

5.3.3 MAVEHA Dataset

Further experimentation has been carried out to investigate the model's performance on the real MAVHEA dataset. As previously outlined in Section 2.4, the 3D ground-truths for this dataset have been corrupted. Furthermore, only 12 keypoints are labelled for each frame in 2D. Thus fine-tuning of the 16 keypoint 2D pose estimation model on this dataset is not possible. Therefore the results presented in this section are produced by a model fine-tuned only on the synthetic infant MINI-RGBD [7] dataset. The inputs to the model have been cropped according to their ground-truth bounding boxes.

The the two model's keypoint definitions, after mapping to the MPII definition, are shown in Fig 5.7. It can be seen that the labelled keypoints for the MAVEHA dataset equate to the limb keypoints of the MINI-RGBD dataset. Therefore quantitative evaluation was carried out for only the limb predictions of the model. As no head or neck keypoints are not present the PCKh score could not be calculated, only the AJPE. This was found to be 13.82 pixels on average.

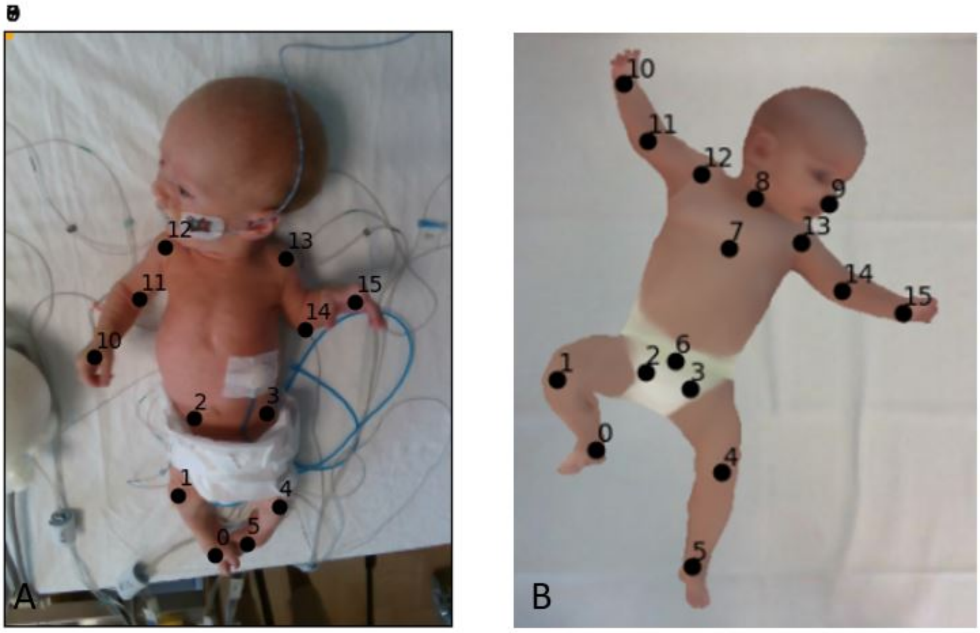

Figure 5.7: Samples from A) MAVHEA dataset and B) MINI-RGBD dataset showing their labelled keypoints. The numbering is in accordance to the MPII definition, see Section 3.4.3. The missing keypoints in A) have been placed at (0,0)

The visualisations of the model's outputs are shown in Fig 5.8. Similar to the other visual outputs presented in this report the inputs increase in complexity from A to E and so both success and failure cases are shown.

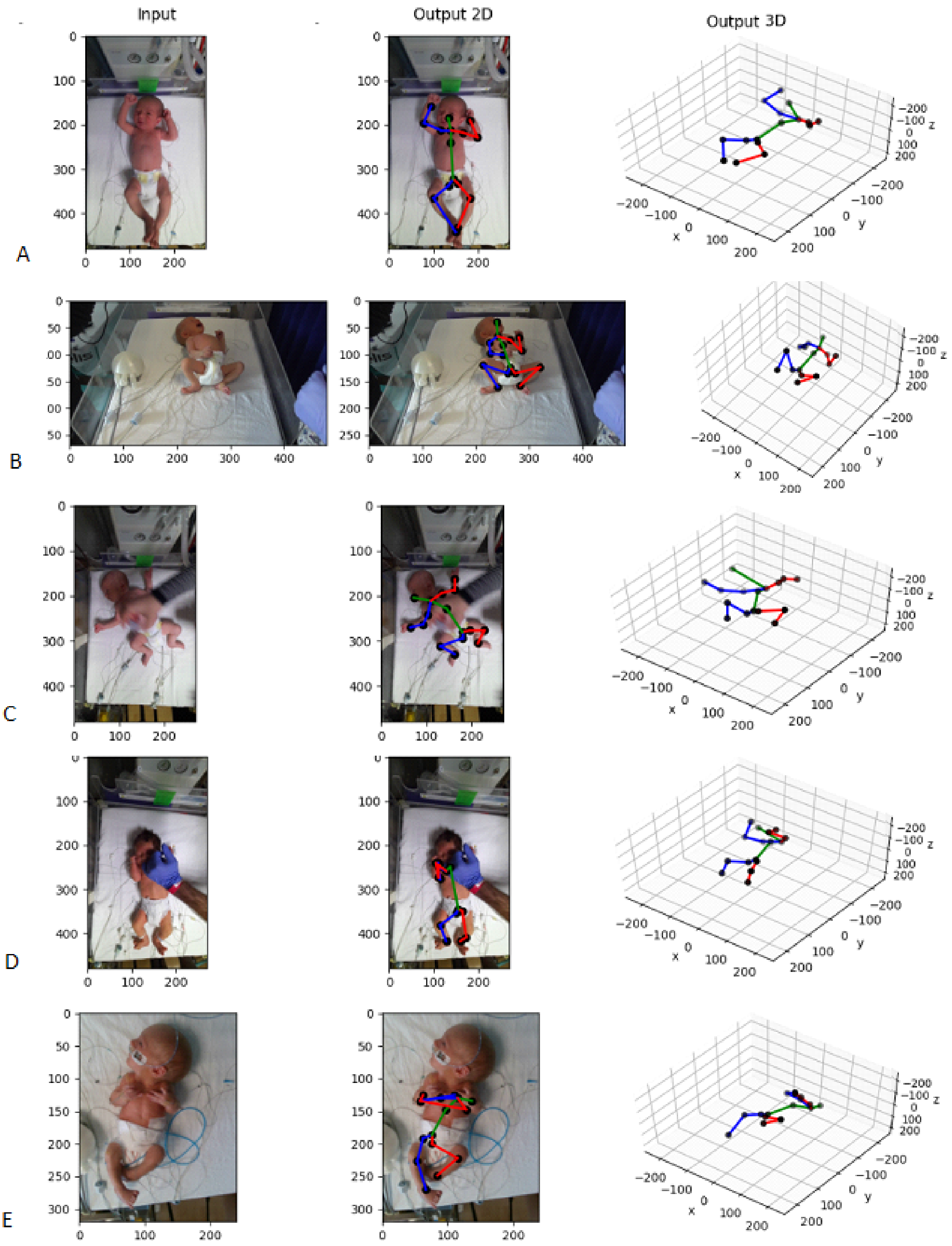

Figure 5.8: A sample of outputs when the model fine-tuned on the synthetic MINI-RGBD dataset [7] is applied to the real MAVEHA dataset.

5.4 Ablative analysis

To better understand the impact of the design choices made in Chapter 3 an ablation study was carried out. The mechanisms of mapping of all datasets keypoints to those defined by the MPII dataset and transfer learning were investigated for both the 2D and 3D models. Additionally for the 2D model the effectiveness of using the Faster R-CNN bounding box to crop the input image was explored. The results of these investigations can be seen in Table 5.2. In all cases, the removal of the aforementioned mechanisms leads to a decrease in the respective model's accuracy.

	PCKh 1.0/%	**AJPE/mm or pixels**
Full 2D Model	93.77	8.17
w/o Adult Pre-training	82.98	10.96
w/o Mapping	79.57	13.36
w/o Bounding Box Crop	62.58	17.41
Full 3D Model	69.28	25.44
w/o Adult Pre-training	52.18	34.18
w/o Mapping	20.19	67.82

Table 5.2: Ablative analysis

Chapter 6

Discussion

This chapter evaluates each of the models developed for this project based on the results presented in Section 5. It then moves on to critically analyse the project's methodology and suggest areas for future work.

6.1 Bounding Box Model

The performance of this model can be characterised as adequate. Further performance from this model could be achieved from techniques, such as transfer learning from the adult domain and an extensive hyper-parameter search. However, in the majority of cases, the discrepancy in the IoU metric was due to an overestimation of the correctly identified infant. This was aided by the padding added ground-truth bounding box generated from the segmentation mask, see Section 3.1.2.

However, there are cases when the bounding box is underestimated, such as Fig 5.1B, and so key-points at the extremities of the infant may be cropped from the model. Although there is evidence that this model can accurately predict out of frame keypoint, such as in Fig 5.2C, this may be one source of performance degradation seen for the outer keypoints such as hands and feet.

The results of the ablation study, presented in Table 5.2, clearly show the large performance gain that adding the bounding box crop mechanism to the 2D pose estimation pre-processing stage leads to. This result was expected due to the self evidently beneficial effects of removing extraneous background objects and standardising of the scale of the subject.

6.2 2D Pose Estimation Model

6.2.1 MPII Dataset

Assessment of the outputs visualised in Fig 5.2 gives insight into the performance and limitations of the model that are not exposed by numeric metrics. Fig 5.2A shows a simple case of a subject straight on to the camera with no complicating factors such as occlusion. The model handles this input well.

More challenging inputs are shown in Fig 5.2B and C in which the legs of the subject are either self-occluded or hidden by another object in the scene. The model also handles this input well. This shows that the model has not only learnt to identify the shapes of keypoints in an image but to infer the location of occluded keypoints from its relationship with visible keypoints and likely human poses. Fig 5.2D provides further proof on the non-trivial inductive power of the model by correctly identifying a keypoint location which is not in frame.

The model does break down however for some highly challenging inputs as can been seen in Fig 5.2E. There are several factors which make this input difficult. Firstly multiple other people are overlapping with the subject meaning that multiple occurrences of certain keypoints will be present in the cropped input image. Furthermore, the pose is highly atypical and there no differentiation in clothing between the incorrectly labelled arm and leg. Pleasingly though the other limbs and central body region are correctly identified.

6.2.2 MINI-RGBD Dataset

Visual comparison of Fig 5.2 and 5.3 shows significant differences between the MPII and MINI-RGBD dataset beyond the obvious change in the subject's proportions. Differences such as the lack of occlusion from objects and the presence of only a single subject in a scene actually simplify the problem. However there are additional challenges to counteract this. For example, the shift from a real to a synthetic dataset is non-trivial. Moreover, there is a tendency in the poses displayed for limbs to cross especially the lower leg. This is exacerbated by the lack of variation in colour or texture between limbs which is supplied in most examples in the MPII training dataset by clothing. These are two of the primary factors that lead to the mislabelling of limbs in the MPII dataset, for example Fig 5.2E.

However, in the most part further training in the infant domain seems to have made the model robust to these challenges as can be seen in Fig 5.3B. This being said these factors can still lead to mislabelling a limited number of very challenging inputs such as Fig 5.3C

The results in Section 5.2.2 shows an increase in several performance metrics between this project's model and the current SOTA [7], such as a 7.23% increase in the PCKh 1.0 evaluation metric. Further comparisons can be made through the analysis of Fig 5.4 which compares the performance of the two models on a keypoint basis. This figure shows that there is a large gain in performance, over 15% PCKh 1.0, on the hip and knee keypoints previously identified area of weakness of the OpenPose model on this dataset [7]. A possible explanation for this is that the OpenPose model is trained on purely adult data. Therefore poses in the lying position with the legs tucked or crossed are rare and hence challenging for the model to accurately predict. However, in the infant domain such as pose is very standard and so further specialised training on such as dataset leads to a specific boost for these limbs.

6.3 3D Lifting Network

The training of the lifting network on the adult MPI_INF_3DHP dataset was successful with reasonable metrics of 64.02mm AJPE and 92.21% PCKh. Similarly, the quantitative results outlined in Section 5.3 show the successful training of the on the synthetic infant data. However observational analysis of 3D poses from the 2D diagrams is an inherently difficult task. Therefore to give insights more nuanced than that in general plausible poses are outputted by the network is difficult and so are not expand on in depth here.

6.3.1 Transfer Learning

However, what can be seen from the comparison of the adult and infant metrics in Fig 5.4 is the relatively poor performance resulting from the domain shift, especially when compared to the results in 2D. The difference in PCKh 1.0 values between the infant and adult models was 27.01% in the 3D case compared to 5.1% in 2D.

There are caveats to this apparent failure though. Firstly the MPI_INF_3DHP metrics were calculated on a validation, not a test set, unlike the MINI-RGBD dataset. Secondly, the threshold used to calculate PCKh is somewhat arbitrary and was designed for use in the adult domain. The original paper presenting the MINI-RGBD dataset [7] recognised this issue stating that the difference in infant proportions means that the head to neck length is much shorter. The solution put forward by Hesse et al. was to arbitrary increase the threshold factor from 0.5 as used in adults to 1 or 2. However, the actual average threshold value used for the MINI-RGBD is still over 5.5 times smaller, 26.6mm compared to 150mm, than that used for the MPI_INF_3DHP dataset. The increased threshold is still pessimistic in it's labelling of correctly identified keypoints. For example in Fig 5.6B, several keypoints including the right elbow are categorised as an incorrect position even though the visual deviation is small to negligible.

Therefore, more important than the comparative PCKh score is to look at the relative patterns between the adult and infant 3D models. It can be seen that the same variation in performance occurs, with keypoints with greater ability to move in relation to the zeroed pelvis joint more difficult to position.

Having said this, for the infant 2D model the PCKh scores were found to be above 90% for both thresholds. Whereas in the 3D although the PCkh 2.0 was still within 5% of the adult PCKh when the threshold was lowered to 1.0 the dropoff was marked. This suggests that the error in the model is not in positioning keypoints in an anatomically implausible position, for instance swapping left and right wrists. Instead, it points to a lack of accuracy the precise location of the keypoint hence the more coarse PCKh 2.0 still scores highly. A key difference between the 2D and 3D case is the camera invariance of the 2D model's predictions. As the 2D model predicts pixel locations it is independent of the camera's intrinsic properties. However, the 3D model is tasked with positioning the infant's keypoints in real space. As part of the solution to this problem, the model must relate differences in pixel co-ordinates to distances in real 3D space. A classical approach to such a task would be to use the intrinsic camera matrix which relates the pixel co-ordinates at a certain depth to real depths based on several camera properties[89]. As the name would suggest this is an individual camera property. Therefore the change in the 3D domain not only contains the challenges common with the 2D case, such as differing subject proportions and poses, but also a changing relationship between pixel and real-world distances. This relationship is essential to make accurate estimations. It may be the case that this subtle fine-tuning is masked by the larger variations in pose and body proportions and so with a small dataset cannot be learnt accurately. The large space between domains coupled with the stringent nature of the metric may explain the poor PCKh 1.0 metric performance.

6.4 End to End model

This section evaluates the performance of the end to end model, when the outputs of the 2D pose estimation model are used as the inputs to the 3D lifting network. This is done for when the model is applied to the MINI-RGBD and MAVHEA infant datasets.

6.4.1 MINI-RGBD Dataset

An observation that can be made from the video of the model's outputs (`https://www.youtube.com/watch?v=rvivVQzxUIc&feature=youtu.be`) is the jittery nature of the estimates. This is not surprising as no account for the temporal relationship between inputs has been made in the model's design. Potential approaches incorporate such information are explored in Section 6.6.

A further notable result was the small decrease in performance when moving from ground-truth to 2D pose estimates as shown in Table 5.1. For comparison, in Martinez et al. [62] found a 60% decrease in the AJPE when using model outputs instead of ground-truth 2D inputs [62], this compares to 12% increase for this project's model. This small performance decrease, therefore, meant the absolute benefit of fine-tuning was also decreased. Especially as Fig 5.6C shows the lifting network cannot rectify incorrectly labelled keypoints. However, even after fine-tuning the performance the decrease between model and ground-truth input was still proportionally lower than for Martinez et al. [62]. This shows the high accuracy of the 2D infant pose estimation model due to the success of the domain shift as well as the arguably easier synthetic domain lacking challenges such as occlusion.

Another positive finding of the project is shown in Table 5.1, which demonstrates the superior performance of the project's model compared to the current SOTA [1] on the MINI-RGBD dataset with a 14.11% increase in the PCKh 1.0 metric. Furthermore, the model implemented in this project only uses an RGB input compared to the previous SOTA method which required an additional depth channel. The implications of this improvement are significant to the potential application and use of infant pose estimation models moving forward. RGB-D sensors are specialised pieces of equipment compared to RGB cameras which are in widespread circulation. Therefore this work moves the capture of infant pose in 3D from a task requiring a specialised setup to one which could potentially run off a smartphone app.

6.4.2 MAVHEA Dataset

Analysis of the performance of the model when applied to the real infant MAVEHA dataset gives insights to the ability of the model fine-tuned on synthetic data to generalise to real data. However this is complicated by the absence of a full set of keypoint ground-truths for this dataset.

Quantitative analysis of the 2D model showed the AJPE error to be 5.65 pixels higher than for the synthetic MINI-RGBD dataset [7]. Interestingly the hip keypoints have the largest error at 24.29 pixels. This diverges from the results of the MINI-RGBD dataset where the hip keypoints were found to be some of the most accurate. The reason for this error is thought to be due the MINI-RGBD dataset defining the hip keypoints on the inside and the MAVEHA dataset on the outside of the infant as can be seen in Fig 5.7.

Further insights can be gained from the visual evaluation of Fig 5.8. Fig 5.8A shows a success case. The 2D output is accurate visually with challenging keypoints such as overlapping ankles correctly identified. Furthermore, a plausible 3D pose is also estimated with features such as the up-stretched right arm correctly identified. It is of note this input has a near-identical overhead camera angle to the MINI-RGBD dataset and self-occlusion is limited. Furthermore, comparing this input to Fig 5.6 it can be seen that the real inputs are much crisper and so differentiation between body parts is visually easier. These factors make Fig 5.8A a relatively simple case, hence, the successful pose estimation.

Success with more challenging inputs can be seen in Fig 5.8B and C. Fig 5.8B's input is at a lower camera angle and also includes limbs overlapping with the body. In the input Fig 5.8C the infant is partially obscured by an adult hand which has obvious feature similarity to the infant's wrists that are being estimated. However, the scale of the adult hand differs by an order of magnitude so differentiation is still possible and the only occluded keypoint is the spine which has a low degree of freedom between the visible pelvis and neck keypoints. Furthermore, the 3D pose that is proposed by the lifting network has a neck to nose segment is visually anomalous. This may be due to the infant being at an angle relative to the camera's axis and augmentation of the 3D training data has not been implemented.

Fig 5.8D shows an input with complete occlusion of the left hand by an adult. In this case, the 2D pose model breaks down and predicts the same keypoints for the left and right arm. Interestingly, the 3D lifting network attempts to fix this error predicting differing right and left arm outputs. This behaviour is not typical as usually a failure of the 2D network results in a 3D failure also, such as in Fig 5.6C. Visual analysis of the accuracy of this prediction by comparing two 2D figures is difficult, however, the 3D pose shown is plausible.

The final output Fig 5.8E is highly challenging with both arms overlapping the infant's body. As with the MINI-RGBD dataset due to the lack of clothing identification of these limbs is difficult. Furthermore, the infant has a nasogastric tube which is not seen in the synthetic training examples. These factors coupled with the angled camera position results in an incorrect 2D and 3D pose estimation of the nose and arm keypoints.

The generation of the outputs in Fig 5.8 also revealed the dependence of the lifting network on the size of the original image. For the 2D network the image is cropped to 256x256 pixels and so the input to the network is independent of the original image size. However, the 2D keypoints are transformed back in the original image's co-ordinate system in the inference post-processing stage. Therefore the magnitude of the inputs to the 3D lifting network is proportional to the size of the input image. This can lead to inputs to the model well outside the domain that the network was trained on and so outputs of incorrect pose and scale. A short term fix to produce Fig 5.8 was to crop and down sample the input image to be approximately equal to that of the MINI-RGBD [7] dataset. A normalisation of the inputs to the lifting network maybe another option to remove this dependence. This would require the retraining of the 3D lifting networks.

6.5 Critical Analysis of Model Design

This section presents a critical analysis of two of the major features in the design the 3D pose estimation model created in this project. Namely the two model design and the mapping of all datasets keypoints to one common framework.

6.5.1 Two Model Design

One of the most obvious decisions in the design of the model network was the division of the task into two instead of a single end to end model. A clear disadvantage to this methodology is the loss of rich visual cues when predicting the 3D keypoint locations. The information carried by 32 integers is less than 256x256 RGB image. However, the work of Martinez et al. [62] and to an extent this project shows 3D poses can be accurately reformed from their 2D poses. This being said the current SOTA adult 3D pose estimation method, MargiPose [64], as described in Section 2.3.2 is an end to end methodology. It also uses the recently proposed differentiable soft-argmax function to calculate keypoint positions from 2D marginal heatmaps. This is in contrast to the argmax function used by the 2D model in this project which is not differentiable [65]. Therefore [64] can backpropagate L2 loss of the final predicted keypoints and so create a truly end to end model unlike the model proposed here.

There are benefits of this approach though. Firstly, the input of the 3D lifting network was simply 32 integers the loading of the data and so training of the network was rapid in comparison to the 2D model. During training on the MINI-RGBD dataset, a 3D epoch with a batch size of 64 was iterated through in 2.19 secs on average compared to 138.84 secs for 2D epoch with a batch size of 16. This fast training time for the 3D lifting network allowed for a hyper-parameter search to be carried out a reasonable amount of time. This benefit was especially clear on the large adult datasets where the training of the 3D network was completed in the order of minutes compared to days for the 2D model.

Secondly, the separation of the 3D and 2D keypoint location allowed for the identification of the proportional error which according to Table 5.1 lies heavily in the 3D domain. This is a departure from the finding of Martinez et al. [62] who found the 2D pose estimation to be the largest source of error. The reasons for the relatively poor performance of the 3D lifting network are explored thoroughly in Section 6.3.

The greatest advantage of this approach, however, is in consideration to the models deployment on in the wild datasets such as the MAVEHA dataset. As previously outlined in Section 2.4, 3D ground truths are difficult to record and so large datasets, especially in the infant domain, do not exist. However, the calculation of 3D keypoints from a simulated model is trivial. The caveat though with such a data source is the inability of a model trained upon it to generalise to in the wild examples due to visual artefacts in the synthetic data. For the 3D lifting network though this is not the case, assuming the data contains infants of correct proportions and relevant poses, as no visual artefacts are present in the input. Therefore fine-tuning may only be required for the 2D model which only requires the relatively easy to ascertain 2D ground truths. Evidence for this is shown in Fig 5.8, in which the breakdown of the synthetically trained model when applied to real data was with the 2D model rather than the 3D lifting network.

6.5.2 Mapping

A major design feature of this model is the mapping of keypoints. The theoretical reasoning behind this mechanism is described in Section 3.4.3. The results presented in Table 5.2 provide evidence for this theory with the improvements to the PCKh 1.0 metric of 14.2% and 49.09% for the 2D and 3D models respectively.

Analysing the results from the 2D model without mapping on a per keypoint basis shows the worst performing areas were for those not present in the adult MPII dataset, for example fingers. However, even when the keypoints not present in the MPII are removed from the metric calculations the PCKh 1.0 was still approximately 5% lower for the mapped model. This provides evidence to the idea that when training the non mapped model the loss function was dominated by the new keypoints. Therefore the fine-tuning required in transfer learning for the other keypoints was suppressed.

The beneficial effect in the 3D case is even greater with a 49.09% increase in the PCKh 1.0 metric. By not mapping the keypoints uniformly the inputs to the infant model have totally different relationships to those pre-learnt in the adult domain. Hence the positive effects of transfer learning are not seen and the model performs very poorly.

There is a disadvantage in the mapping as outlined and implemented in this project though. The dataset which is mapped to is determined by which source has the fewest keypoints. In this project that was the MPII dataset with 16. Therefore all other models including the final infant model predicted 16 keypoints. This meant that 9 keypoints for which ground-truths were available for are not estimated on the infant dataset. The MPII dataset was chosen at the outset of this project due to its wide use in academia including previously by the MAVEHA project. However on reflection, a different 2D pose dataset would now be selected such as the RGB images from the MPI_INF_3DHP dataset. This could allow more keypoints to be estimated for MINI-RGBD inputs and so the model could give a more detailed description of the infant's pose.

6.6 Recommendations for Future Research

The model outlined in this report performs the task of 3D infant pose estimation to a SOTA accuracy. However, the work outlined here has several limitations including the limited number of keypoints estimated due to dataset choices and the challenge of the 3D domain shift due to changing cameras. Furthermore, the model's functionality is not fully tested and so limited by challenging inputs such as varying camera angle and out of frame limbs. All of which are valid areas for further research.

However, the largest outstanding research challenge is the generalisation of the model presented here to real infant datasets, such as MAVHEA. The analysis of 5.8 shows that the model can generalise to real data for inputs with similar to those seen in the synthetic training set. However, the real MAVHEA dataset has a number of deviations from the synthetic data such as occlusion by adults, medical equipment, rotation of the infant and differing camera angles. Furthermore, it was found that the 3D lifting network was sensitive to changes in the input image size. It is reasonable to assume that fine-tuning on a real dataset would allow for adaption to these changes in domain and so result in an increase in the accuracy of estimations made by the model in the wild. However, this is inhibited by the availability of ground-truths.

The MAVHEA dataset does have 2D ground truths but only 12 are available compared to the 16 predicted by this model. A solution to this problem would be to redefine the common keypoint definition as the MAVHEA dataset. This would require the re-mapping of the MPII, MPI_INF_3DHP and MINI-RGBD accordingly followed by the retraining of each of their respective models. The cost of this approach would be the loss of 4 keypoints in the expression of an infant's pose.

The availability of 3D ground truths is trickier. As outlined in Section 6.4.2, there is an argument that the necessity for such a process is less than in the 2D case due to the two model design used in this project. However, the current SOTA model in the adult domain, MargiPose [64], is an end-to-end methodology using 2D marginal heatmaps and the soft arg-max function. Such a technique has been shown to outperform the approaches used in this project in the adult domain and so would be expected to also perform similarly in the infant domain. As this project has developed methods to use transferring to shift deep learning models from the adult and infant domain, if the 3D ground-truths for a real infant dataset became available the research into the development of a model based on MargiPose [64] would be logical.

If the 3D ground-truths remain corrupted however, the fine-tuning of the 3D pose lifting network maybe possible with only the 2D data the research into the adaption of the work of Chen et al. [67], outlined in Section 2.3.2. This approach uses self-constancy constraints of the lifting network to form a loss function. Furthermore, this work increases the accuracy of the model by incorporating the output of a temporal discriminator trained using a GAN [68] into the training loss function. Research into the addition of such a model could reduce the unstable temporal output of the model.

6.7 Summary

This chapter has explored the relative success and failures of the models developed in this project. It has found the bounding box model developed to have an adequate level of performance. It was further noted is the significant increase in performance that cropping the inputs to the 2D infant pose estimation according to this model has on performance.

The factors leading to failure outputs of the 2D pose estimation model, such as self-occlusion and lack of limb differentiation, were identified as well as the differences in the adult and infant 2D dataset. Also, the additional fine-tuning to the infant dataset was concluded to be a major factor in this project's superior performance compared to the current SOTA 2D infant pose estimation model.

During the evaluation of the 3D model, the drop off in performance of the infant compared to adult model was investigated in detail. Several explanations were put forward including the failure of the model to fine-tune its output to a camera with different intrinsic properties due to the other larger changes within the domain including the subjects typical pose and body proportions. The relatively small deviation in performance when inputs to the model were shifted from ground-truths to 2D pose estimates was attributed to the high performance of the 2D model.

The crucial point of differentiation with the current 3D pose estimation SOTA was not only the increase in performance but also the removal of the depth channel from the input. This allows the capture of data from a much wider and more ubiquitous range of cameras, such as a smartphone, than previously possible.

Critical analysis of the model's design identified a clear disadvantage of the two model design being that the lifting network loses the visual cues of the original RGB input. However, the advantages of the fast training time and potential generalisation to the real dataset were also acknowledged. Further analysis looked at the mapping methodology. This was shown to be essential to the transfer learning approach, especially for the 3D lifting network. A major limitation was the reduction in keypoints that could be predicted in the infant dataset due to the need to be mapped to the 16 keypoint MPII dataset.

Finally, recommendations for future research work were presented. Improvements to the current model based on the limitations previously outlined were explored. However, the largest research challenge moving forward is the successful generalisation of this work to real infant data for which the 3D ground-truths scarce. Two approaches [64, 67] for future research were proposed to aid the performance of the model in the wild which represent the current SOTA 3D supervised and unsupervised adult pose estimation techniques.

Chapter 7

Conclusion

The aim of this project was to extend the scientific literature by investigating if deep learning techniques developed in the 3D adult pose estimation domain could be successfully adapted to the problem of 3D infant pose estimation via transfer learning.

A two model design consisting of a 2D infant pose estimation model and a 3D lifting network was successfully implemented. The model architectures were inspired by those proposed by Xiao et al. [8] and Martinez et al. [62]. These were pre-trained on large adult datasets and through the use of transfer learning fine-tuned to smaller infant datasets.

The resulting model outperformed the previous 2D and 3D SOTA methodologies on the MINI-RGBD dataset [7] with AJPE scores of 8.17 pixels and 28.47mm for the 2D and 3D models respectively. Furthermore, the 3D model takes RGB images as input and so does not require a depth channel, unlike previous work.

There are limitations to the model including the restriction on the number of keypoints that can be estimated due to the mapping technique and the selection of adult datasets. Also, the relatively poor performance of the 3D lifting network, seen in the infant but not the adult domain, is an area for further research.

However, the most pressing area for future research is the development of the models ability to generalise to real data sources. The current lack of 3D ground-truths for real infant datasets presents a challenge to this process. The work of Chen et al. [67] and the MargiPose model [64] have been proposed as approaches to be investigated in future research depending on how the availability of 3D ground-truths develops.

In conclusion, the work reported here has shown that SOTA 3D infant pose estimation can be produced via the adaption of adult pose estimation techniques using transfer learning. Moreover, these models do not require a depth input. Therefore the input to such models can be captured by the common RGB video camera. Such devices are now ubiquitous in the modern smartphone era. The wider implication of this work is therefore that infant tracking systems that previously required large specialised equipment setups now require only a video camera.

This project provides the initial technology required to develop an Automated General Movement Assessment that is run off a simple smartphone app by a non-technical operator. Such a development could revolutionise the diagnosis of a wide range of disorders linked to abnormal infant movements due to the numerous possible deployment settings and the low levels of training required to make an initial diagnosis. This could significantly lower the average age of diagnosis and consequentially improve the clinically management and outcomes for infants born with neurological disorders on a worldwide basis.

Bibliography

[1] Nikolas Hesse, Gregor Stachowiak, Timo Breuer, and Michael Arens. Estimating body pose of infants in depth images using random ferns. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 35–43, 2015. pages 1, 7, 16, 40, 48

[2] Openpose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation. `https://github.com/CMU-Perceptual-Computing-Lab/openpose`. Accessed: 27-05-2020. pages 1, 8, 38

[3] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. pages 1, 9, 11, 16, 18

[4] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. pages 1, 23

[5] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017. pages 1, 5, 11, 12, 14, 15, 17, 24, 39

[6] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. pages 1, 12, 15, 17, 23, 27

[7] Nikolas Hesse, Christoph Bodensteiner, Michael Arens, Ulrich G Hofmann, Raphael Weinberger, and A Sebastian Schroeder. Computer vision for medical infant motion analysis: State of the art and rgb-d data set. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. pages 1, 4, 8, 15, 17, 23, 24, 31, 32, 38, 39, 41, 42, 43, 46, 47, 49, 50, 55

[8] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer*

vision (ECCV), pages 466–481, 2018. pages 1, 10, 15, 17, 20, 21, 22, 25, 27, 32, 35, 55

[9] Midori Kitagawa and Brian Windsor. *MoCap for artists: workflow and techniques for motion capture*. CRC Press, 2012. pages 1

[10] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. IEEE, 2009. pages 2, 5

[11] Krystal Rose Higgins, Eric J Farraro, John Tapley, Kumaresan Manickavelu, and Saurav Mukherjee. Virtual dressing room, February 20 2018. US Patent 9,898,742. pages 2, 5

[12] Philip Teitelbaum, Osnat Teitelbaum, J Nye, Joshua B. Fryman, and Ralph G. Maurer. Movement analysis in infancy may be useful for early diagnosis of autism. *Proceedings of the National Academy of Sciences of the United States of America*, 95 23:13982–7, 1998. pages 3

[13] Heinz F Prechtl. Qualitative changes of spontaneous movements in fetus and preterm infant are a marker of neurological dysfunction. *Early human development*, 1990. pages 3

[14] Mijna Hadders-Algra. Early diagnosis and early intervention in cerebral palsy. *Frontiers in neurology*, 5:185, 09 2014. doi: 10.3389/fneur.2014.00185. pages 3

[15] K Himmelmann and P Uvebrant. The panorama of cerebral palsy in sweden part xii shows that patterns changed in the birth years 2007–2010. *Acta Paediatrica*, 107(3):462–468, 2018. pages 3

[16] Alicia Spittle, Jane Orton, Peter J Anderson, Roslyn Boyd, and Lex W Doyle. Early developmental intervention programmes provided post hospital discharge to prevent motor and cognitive impairment in preterm infants. *Cochrane Database of Systematic Reviews*, (11), 2015. pages 3

[17] Betty Hutchon, Deanna Gibbs, Phillip Harniess, Sally Jary, Siew-Lian Crossley, Jane V Moffat, Neela Basu, and Anna P Basu. Early intervention programmes for infants at high risk of atypical neurodevelopmental outcome. *Developmental Medicine & Child Neurology*, 61(12):1362–1367, 2019. pages 3

[18] Schirin Akhbari Ziegler, Tineke Dirks, and Mijna Hadders-Algra. Coaching in early physical therapy intervention: the copca program as an example of translation of theory into practice. *Disability and rehabilitation*, 41(15):1846–1854, 2019. pages 3

[19] Gija Rackauskaite, Jakob Granild-Jensen, Esben Flachs, and Peter Uldall. Predictors for early diagnosis of cerebral palsy from the danish national registry data: Po99-98363. *Developmental Medicine & Child Neurology*, 57, 2015. pages 3

[20] Mijna Hadders-Algra. General movements: a window for early identification of children at high risk for developmental disorders. *The Journal of pediatrics*, 145(2):S12–S18, 2004. pages 3, 4

[21] Mijna Hadders-Algra, Annelies MC Mavinkurve-Groothuis, Sabina E Groen, Elisabeth F Stremmelaar, Albert Martijn, and Phillipa R Butcher. Quality of general movements and the development of minor neurological dysfunction at toddler and school age. *Clinical Rehabilitation*, 18(3):287–299, 2004. pages 4

[22] Claire Marcroft, Aftab Khan, Nicholas D Embleton, Michael Trenell, and Thomas Plötz. Movement recognition technology as a method of assessing spontaneous general movements in high risk infants. *Frontiers in neurology*, 5: 284, 2015. pages 4

[23] Eileen Ricci, Christa Einspieler, and Alexa K Craig. Feasibility of using the general movements assessment of infants in the united states. *Physical & occupational therapy in pediatrics*, 38(3):269–279, 2018. pages 4

[24] A Sebastian Schroeder, Nikolas Hesse, Raphael Weinberger, Uta Tacke, Lucia Gerstl, Anne Hilgendorff, Florian Heinen, Michael Arens, Linze J Dijkstra, Sergi Pujades Rocamora, et al. General movement assessment from videos of computed 3d infant body models is equally effective compared to conventional rgb video rating. *Early Human Development*, 144:104967, 2020. pages 5

[25] K. D. McCay, E. S. L. Ho, H. P. H. Shum, G. Fehringer, C. Marcroft, and N. D. Embleton. Abnormal infant movements classification with deep learning on pose-based features. *IEEE Access*, 8:51582–51592, 2020. pages 5

[26] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. pages 5

[27] Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J Black, Christoph Bodensteiner, Michael Arens, Ulrich G Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, et al. Learning an infant body model from rgb-d data for accurate full body motion analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 792–800. Springer, 2018. pages 5, 8

[28] Azure kinect dk documentation. `https://docs.microsoft.com/en-gb/azure/Kinect-dk/`. Accessed: 26-05-2020. pages 5, 6, 8, 15

[29] Dominik Karch, Keun-Sun Kim, Katarzyna Wochner, Joachim Pietz, Hartmut Dickhaus, and Heike Philippi. Quantification of the segmental kinematics of spontaneous infant movements. *Journal of biomechanics*, 41(13):2860–2867, 2008. pages 5

[30] Dominik Karch, Keun-Sun Kang, Katarzyna Wochner, Heike Philippi, Mijna Hadders-Algra, Joachim Pietz, and Hartmut Dickhaus. Kinematic assessment of stereotypy in spontaneous movements in infants. *Gait & posture*, 36(2): 307–311, 2012. pages 5

[31] Franziska Heinze, Katharina Hesels, Nico Breitbach-Faller, Thomas Schmitz-Rode, and Catherine Disselhorst-Klug. Movement analysis by accelerometry of newborns and infants for the early detection of movement disorders due to infantile cerebral palsy. *Medical & biological engineering & computing*, 48(8): 765–772, 2010. pages 5

[32] D Gravem, M Singh, C Chen, J Rich, J Vaughan, K Goldberg, F Waffarn, P Chou, D Cooper, D Reinkensmeyer, et al. Assessment of infant movement with a compact wireless accelerometer system. *Journal of Medical Devices*, 6(2), 2012. pages 5

[33] Andraž Rihar, Matjaž Mihelj, Jure Pašič, Janko Kolar, and Marko Munih. Infant trunk posture and arm movement assessment using pressure mattress, inertial and magnetic measurement units (imus). *Journal of neuroengineering and rehabilitation*, 11(1):133, 2014. pages 5

[34] Crystal Jiang, Christianne J Lane, Emily Perkins, Derek Schiesel, and Beth A Smith. Determining if wearable sensors affect infant leg movement frequency. *Developmental neurorehabilitation*, 21(2):133–136, 2018. pages 5

[35] Lars Adde, Jorunn L Helbostad, Alexander Refsum Jensenius, Gunnar Taraldsen, and Ragnhild Støen. Using computer-based video analysis in the study of fidgety movements. *Early human development*, 85(9):541–547, 2009. pages 6, 7

[36] Annette Stahl, Christian Schellewald, Øyvind Stavdahl, Ole Morten Aamo, Lars Adde, and Harald Kirkerod. An optical flow-based method to predict infantile cerebral palsy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(4):605–614, 2012. pages 6

[37] Hodjat Rahmati, Ralf Dragon, Ole Morten Aamo, Lars Adde, Øyvind Stavdahl, and Luc Van Gool. Weakly supervised motion segmentation with particle matching. *Computer Vision and Image Understanding*, 140:30–42, 2015. pages 6

[38] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187 – 1200, Jun 2014. URL `http://lmb.informatik.uni-freiburg.de/Publications/2014/OB14b`. Preprint. pages 6

[39] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013. pages 6

[40] Mikkel Damgaard Olsen, Anna Herskind, Jens Bo Nielsen, and Rasmus R Paulsen. Body part tracking of infants. In *2014 22nd International Conference on Pattern Recognition*, pages 2167–2172. IEEE, 2014. pages 7

[41] Mikkel Damgaard Olsen, Anna Herskind, Jens Bo Nielsen, and Rasmus Reinhold Paulsen. Model-based motion tracking of infants. In *European Conference on Computer Vision*, pages 673–685. Springer, 2014. pages 7, 8

[42] Ananth Ranganathan. The levenberg-marquardt algorithm. *Tutoral on LM algorithm*, 11(1):101–110, 2004. pages 7

[43] Annalisa Cenci, Daniele Liciotti, Emanuele Frontoni, Primo Zingaretti, and Virgilio Paolo Carnielli. Movements analysis of preterm infants by using depth sensor. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, pages 1–9, 2017. pages 7

[44] Shreyas S Shivakumar, Helen Loeb, Daniel K Bogen, Frances Shofer, Phillip Bryant, Laura Prosser, and Michelle J Johnson. Stereo 3d tracking of infants in natural play conditions. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 841–846. IEEE, 2017. pages 7

[45] Thomas Helten, Andreas Baak, Gaurav Bharaj, Meinard Müller, Hans-Peter Seidel, and Christian Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *2013 International Conference on 3D Vision-3DV 2013*, pages 279–286. IEEE, 2013. pages 8

[46] Nikolas Hesse, A Sebastian Schröder, Wolfgang Müller-Felber, Christoph Bodensteiner, Michael Arens, and Ulrich G Hofmann. Body pose estimation in depth images for infant motion analysis. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1909–1912. IEEE, 2017. pages 8

[47] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. pages 8

[48] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. pages 8

[49] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998. pages 9

[50] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015. pages 9, 15

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. pages 9, 18

[52] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015. pages 9, 10

[53] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. pages 10, 13

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. pages 10, 12, 15, 21, 22, 23

[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. pages 10, 20, 23, 32

[56] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. pages 10, 11

[57] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014. pages 11

[58] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. pages 12

[59] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. pages 12

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. pages 12

[61] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017. pages 12, 13

[62] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. pages 12, 17, 23, 24, 25, 27, 32, 33, 48, 50, 51, 55

[63] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. pages 13

[64] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1477–1485. IEEE, 2019. pages 13, 14, 50, 52, 53, 54, 55

[65] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010. pages 13, 50

[66] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018. pages 13

[67] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2019. pages 14, 53, 54, 55

[68] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. pages 14, 53

[69] K Sreedhar Reddy and Dr K Rama Linga Reddy. Enlargement of image based upon interpolation techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12):4631, 2013. pages 18

[70] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. pages 19

[71] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. pages 19

[72] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. pages 19

[73] Marcel Berger. *Geometry i*. Springer Science & Business Media, 2009. pages 20

[74] torchvision.models¶, . URL https://pytorch.org/docs/stable/torchvision/models.html. pages 20, 23, 32

[75] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. pages 20, 23, 24

[76] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. pages 20

[77] Alexei Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*, 2018. pages 23, 24, 34

[78] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. pages 23

[79] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. pages 24

[80] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. pages 24

[81] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019. pages 24

[82] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007. pages 25

[83] Pytorch. https://pytorch.org/, . Accessed: 29-05-2020. pages 29

[84] URL https://numpy.org/. pages 29

[85] Visualization with python¶. URL https://matplotlib.org/. pages 29, 31

[86] Aug 2020. URL https://opencv.org/. pages 29

[87] Sasank Chilamkurthy. Writing custom datasets, dataloaders and transforms¶. URL https://pytorch.org/tutorials/beginner/data_loading_tutorial.html. pages 29, 30, 31, 34

[88] Dataloader. URL https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader. pages 29, 31, 32

[89] Linda Shapiro and George Stockman. *Chapter 13 3D Sensing*, page 14–17. Michigan State, 1999. pages 48