Master's Thesis

Artificial Intelligence

Exploring adult biases in child pose estimation

By:

Vincent Brouwers
6971741
Utrecht University
Utrecht, The Netherlands

Supervised by:

Project Supervisor / First Examiner
Ronald Poppe
Utrecht University
Utrecht, The Netherlands

Daily Supervisor
Metehan Doyran
Utrecht University
Utrecht, The Netherlands

Second Examiner
Albert Salah
Boğaziçi University
Istanbul, Turkey

Submitted: August 18th, 2021

Acknowledgements

This thesis is the result of a long process that would not have been possible without the support of the people around me. To start off, a big thanks to Ronald Poppe, Metehan Doyren, and Albert for supervising me. I am very grateful for all the guidance and help you gave me. Especially thank you for continuously providing such valuable feedback on the many, many iterations of this document. Thank you to Priyanka Singhvi for having my back for the last few weeks and also for helping me label my data. Also a special thank you to my project partner Feyi Olalere with whom it was a pleasure to work so closely.

Thank you to my roommates Bram and Lennert who put up with my "irregular" work schedule and who reminded me to take a break ever so often. And last but not least, of course also a big thank you to my family for their unwavering support, even though I may not have visited that often in the last few months.

Abstract

In recent years, Human Pose Estimation (HPE) algorithms have become increasingly well-performing in localizing the joint locations of humans from images. Besides benefitting from the fast-paced innovations in the field of deep-learning, these models benefit from large-scale manually labeled HPE datasets. These datasets, however, consist mostly of annotations for adult people, whilst underrepresenting children. As children go through a considerable change in body structure throughout puberty, there are several distinct anatomical differences between prepubescent children and adults. This provides reason to believe there to be a performance regression when State Of The Art HPE models are tested on children.

We experimentally demonstrate that modern pose estimators indeed struggle comparatively more with estimating child poses than the poses of adults. We furthermore finetune a benchmark HPE model on child data to verify if this performance difference is due to data limitation or due to model limitations. This is done using a newly collected child-specific dataset that we dub *Kinetikids-pose*. This experiment, however, did not culminate in a conclusive result.

Kinetikids-pose is compiled from photos and video frames of children performing sporting activities. It is to our knowledge the first monocular child HPE dataset that is publicly available. We also present and share two filtered subsets of the COCO validation split: COCO Adult and COCO Child. These are, as the names suggest, subsets filtered to contain either solely adults or solely children.

Contents

Chapter 1

Introduction

1.1 Motivation

In current times, deep learning models are becoming increasingly more capable of understanding the world around us. Computer vision models can detect and locate thousands of different objects in images [76] and natural language models are becoming increasingly performant in condensing internet-scale information into their neurons [7]

Another field where deep learning models are extensively used, HPE, has not yet achieved the same level of success. An HPE model attempts to localize specific human joints from image data, to construct a digital representation of the pose of a subject. Currently, even the best-performing HPE models are not yet performant enough for many practical applications [18]. We suspect this to be especially true for applications that revolve around children, which are underrepresented in most datasets these models are trained on [83].

Where there are many large and varied datasets for visual objects or text, such datasets are less prevalent for HPE. Moreover, through the way these datasets are constructed, the data is often collected with a focus on adult activities, causing samples of children to be underrepresented. We suggest this makes it difficult to train models on them that generalize well to children.

This thesis aims to understand the effects of the biases in these datasets regarding their generalization on children. We also introduce a new dataset called *Kinetikids*, which we construct to test our hypotheses. We show the performance of current State Of The Art (SOTA) pose estimation models on this dataset, both with and without finetuning on it.

1.2 Pose Estimation

HPE is the study of estimating the location of skeletal keypoints of a person in an image or video. The derived pose can be used in a multitude of applications, such as to animate a digital character [18]. The estimations can also be used as part of a larger pipeline, such as for action recognition [91]. On children, applications include behavioral studies and early identification of autism [77, 65] or cerebral palsy [36].

The focus of this thesis will be on HPE via monocular images, such as standard photographs or videos. Using specialized hardware, it is also possible to perform HPE on binocolar [68], or RGB-Depth (RGB-D) [36] images. This can provide useful additional 3D information, but relies on said specialized hardware.

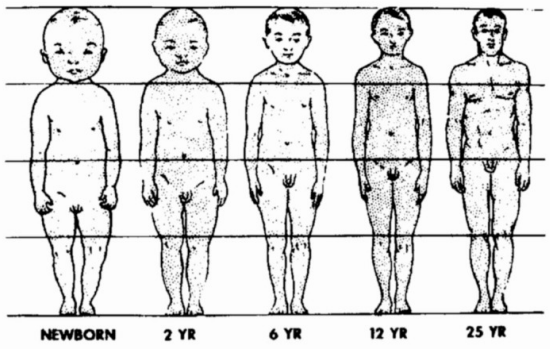
1.3 Children are not miniature adults

HPE algorithms require some form of internal understanding of the human anatomy to work. In certain cases this is explicitly implemented as part of the algorithm [27, 48], but often this understanding is implicitly learned [89, 88, 57]. Because most research into pose estimation is

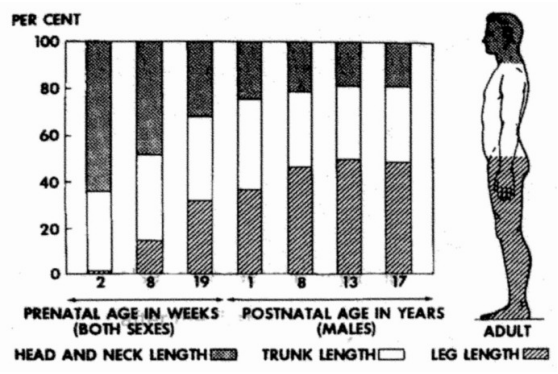
carried out with datasets consisting of mostly adults, these algorithms mainly learned to understand the anatomy of said adults. This knowledge, however, is not directly transferrable to the anatomy of children.

Children are not just smaller versions of human adults, as human bodily proportions continuously change from infancy to adulthood [38]. As visualized in Figure 1.1a and Figure 1.1b, infant bodies have relatively large head and trunk sizes compared to their small neck and legs. More specifically: the head of the average infant is about one-fourth of its total size, whereas that of an adult is about one-seventh of its size [83]. Meanwhile, the proportion that the neck and head take up combined only decreases marginally. They combine to be just over one-fourth for a newborn, whereas the head and neck of an adult take up just one-fifth of their body, proportionally. Lastly, the upper limbs of infants are longer than their lower limbs; this is the opposite in adults.

These differences in structure make it so that one cannot take the posture of a child as equivalent to that of a miniature adult. Models with internal knowledge about human anatomy thus need to adapt to this difference.



(a) Schematic drawings of the human anatomy throughout its development.



(b) Chart graphing the proportions of head and neck, trunk and leg lengths throughout a human's development

Figure 1.1: Visualization of changes in bodily proportions from infant to adult. From "An Overview of Anatomical Considerations of Infants and Children in the Adult World of Automobile Safety Design"[38]

1.4 Thesis Goals

This thesis will explore the effect of the adult biases in current datasets on the generalizability of HPE models on children. By developing a novel child-centered pose estimation dataset, the effect of these biases in existing datasets can be exposed. This dataset also enables further research into HPE on children.

The contents of this thesis are built around the following research question:

> "Is the performance of current SOTA pose estimation on children limited by the adult-biases of the datasets that they are trained on?"

To answer this main research question, the following sub-research questions are composed:

RQ 1. Is there a difference in performance between pose estimation on children when compared to adults?

We know that there are *a)* anatomical differences between adults and children, that *b)* children are underrepresented in current HPE datasets, and that *c)* machine learning models are less performant on out-of-domain data. We thus hypothesize that there is indeed a performance degradation when these models are applied to child subjects.

It stands to reason that some approaches are more robust to anatomical changes than others. We thus also test if some approaches in pose estimation suffer a larger performance degradation than others.

RQ 2. Do SOTA models improve their accuracy on children when trained with a child-specific dataset?

Using part of a dataset to train an HPE model would logically yield an increase in accuracy on that dataset. We will thus examine if training on child data from one dataset also increases the accuracy on children from other datasets.

1.5 Thesis Contribution

color=orange, author=Vincent Brouwers (dev comment)]TODO: Make sure the contributions tie in with the research questions. While there are earlier attempts at creating child-focussed datasets for pose estimation [83], none are sufficiently large to be compared with large-scale datasets such as J-HMDB [42] or MPII [3]. Other attempts also focus on pose estimation in one specific domain, using depth cameras [35, 36]. To our knowledge, ours is the first public HPE dataset with a focus on children.

This thesis contributes to the academic community:

- A new, publicly available, monocular joint-annotated video dataset of expressive children.
- An extensive evaluation of adult bias in current HPE datasets.

1.6 Outline

This thesis will continue by addressing and discussing techniques and theories related to this project in Chapter 2. Chapter 3 then explains the methodology of our approach in detail. The results of this method will be described in Chapter 4, which will be further discussed and analyzed in Chapter 5. Closing the thesis, Chapter 6 states out conclusions to the stated research questions.

Chapter 2

Related Works

This chapter will introduce and explain previous works related to this thesis. The first section (2.1) will introduce the various methods and approaches of 2D pose estimation. Following this, the second section (2.3) lists and summarizes current HPE datasets. Finally, the third section (2.4) looks into previous works on HPE on children.

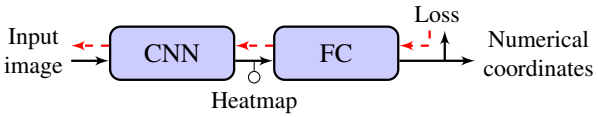
2.1 Overview of Pose Estimation

Human Pose Estimation is the study of localizing joint and pose information from images or video. For the rest of this thesis, it will specifically refer to 2D localization via monocular images. To estimate the joint positions of a human, a model needs to output the correct coordinates for each joint on an image. As demonstrated in Figure 2.1, these estimated "Point of Intersets (POIs)" can be connected to form a skeleton representation of the limbs of the subject. This process of localizing POIs is also called "keypoint detection", and is an important aspect of various other subfields of computer vision. such as face recognition [84, 1] or camera stabilization[47].

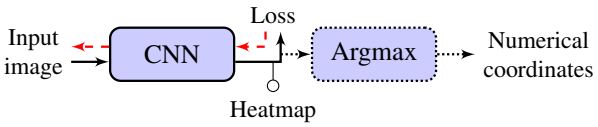
Figure 2.1: Example of how pose estimation applied to multiple athletes in an image. mage and annotations extracted from PoseTrack [4].

2.1.1 Localizing Keypoints

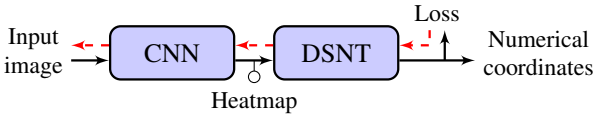
An intuitive approach for estimating the keypoint coordinates would be to use a regressive model to predict them directly from the image data. This method was used for in earlier attempts at for pose estimation with Neural Network (NN) [89, 64], but got surpassed as SOTA by a more performant heatmap-based approaches [63, 19, 64]. More recently, the regressive approaches are making a comeback with the introduction of the *SoftArgMax* function [59] and the similar DSNT layer [67]. We discuss these three classes of approach in the subsequent paragraphs, see Figure 2.2 for a summarized graphical explanation.



(a) Coordinate regression with fully connected layers



(b) Joint detection via heatmap prediction



(c) Coordinate regression via differentiable argmax (DSNT[67] depicted)

Figure 2.2: The arrows indicate inference during training (black, solid), during testing (black, dotted) and the gradient flow (dashed red). From Nibali et al. [67].

"Classical" regression-based

Regression-based approaches attempt to directly predict the x and y coordinates of keypoints through a regression model. Ever since AlexNet showed the might of Convolutional Neural Networks (CNNs) for classifying images [50], it became clear that these types of models were capable tools for computer vision. It was thus only a matter of time before the first CNNs were used to take on HPE. DeepPose by Google's Toshev and Szegedy [89] was, to our knowledge, the first CNN model that was created specifically for pose estimation. They used multiple AlexNet-derived [50] regression models to initially predict a coarse absolute localization, followed by multiple stages of refinement. The refinement stages were a result of the size limitations of models of the time. Their model could only use images of 220×220 in resolution, limiting the level of detail it could work with. The subsequent refinement models only needed to focus on a crop around the earlier prediction, allowing them to look at this region in a relatively higher resolution.

From then on, many methods followed, though currently few score a competitive performance [87]. This is believed to be because the regression from 2D spatial data to a single coordinate is a difficult function to approximate [18, 88].

CNNs are by design spatially invariant, meaning its abilities to recognize a pattern are equal regardless of the spatial location of the pattern. For numerical regression, however, one needs to transform pixel values to coordinates. This is a non-trivial function, as the pixel values themselves contain no information about their spatial location.

Figure 2.3: Diagram of the DeepPose model[89]. The blue blocks represent convolutional layers, whereas the green blocks represent fully connected layers. The *regressor* network outputs absolute coordinates, which are combined with refinement deltas outputted by multiple stages of *refiner* networks. These networks receive a localized crop around the initial prediction.

Detection-based

Detection-based algorithms partition the image into sections and attempt to predict the likelihood that a keypoint is located in that section [18]. A form of the detection-based approach, heatmap-based, is used by most current SOTA models [87].

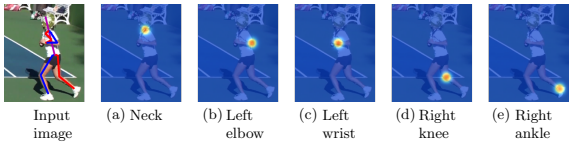
The main advantage of these techniques over the regression-based approaches is that they do not have to learn the complex nonlinear regression function; these models do not produce continuous x and y values as keypoint coordinates. Instead, they produce heatmaps for each keypoint that highlight the most probable location for each type of joint. First used by by Tompson et al. [88] on HPE, this is still a widely used method [18]. To convert the heatmaps to coordinates, an *argmax* post-processing step is used. This retrieves the final x and y coordinates from the most likely locations of each keypoint during inference.

This reliance on the *argmax*-function for inference, however, is also one of the main drawbacks of this approach [18, 64, 87]. Due to the non-differentiable nature of the *argmax* function, these models cannot be trained end-to-end. Instead, they require artificially crafted ground-truth heatmaps for their supervised training process. These ground-truth heatmaps are generated by placing a 2D Gaussian kernel (or less commonly a Bernoulli kernel [37]) on the location of the ground-truth coordinates. The model is then trained to produce heatmaps that resemble these [18].

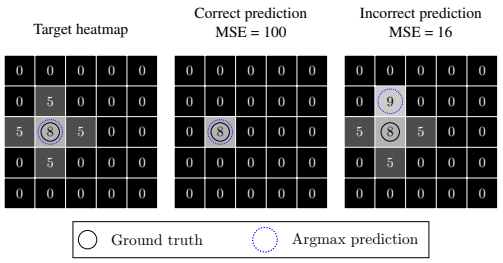
The reliance of the *argmax* function also induces a disconnect between the training objective and the actual objective of predicting coordinates during inference [67]. During inference, only the brightest pixel of a heatmap contributes to a prediction, while during training all pixels contribute to the heatmap similarity loss. This can have the unintended consequence where an objectively better prediction results in a higher loss than a lesser prediction, as illustrated in Figure 2.4b.

Furthermore, these models also suffer from quantization issues during inference. Where the regression-based approaches can estimate coordinates on a continuous scale, these detection-based approaches cannot. The *argmax* function can only produce integer values in the range of $[0, n-1]$ for an output dimension of size n. This, in turn, restricts these methods to discrete values, bound by the resolution of the heatmap [18, 67, 87].

This last problem, also known as quantization, can be mitigated by producing a heatmap of the same resolution as the input image, which is employed by many of the top-scoring methods [66, 92, 98, 8]. This, however, has its own drawbacks as this demands more storage, computation, and memory.



(a) Example heatmaps, as produced by a stacked hourglass model. Each heatmap indicates the predicted likelihood of one type of joint being in the heatmap. From Newell, Yang, and Deng [66].



(b) When heatmap matching, it is possible for predictions to worsen despite the pixel-wise MSE improving. Caption and images from Nibali et al. [67].

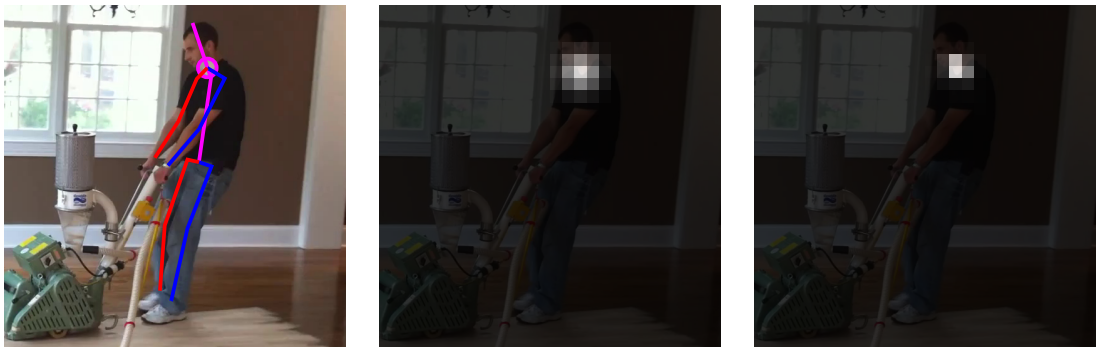
Figure 2.4

Regression through part-detection

Regression through part-detection is a recent development where a differentiable approximation to the *argmax* function is utilized to unify the two aforementioned approaches. This, in many ways, combines the best of both worlds. It allows for networks that are both end-to-end trainable (as the regressive models), as well as being spatially invariant (such as the heatmap models).

In 2017, Luvizon, Tabia, and Picard [59] showed how the differentiable *soft-argmax* function introduced by Finn et al. [28], can directly convert heatmaps into keypoint coordinates. They showed how their novel regression-based model outperformed both normal regression and heatmap approaches when used with a similar architecture. Parallel to Luvizon, Tabia, and Picard, [67] developed the Differentiable Spatial to Numerical Transform (DSNT), which served a similar purpose. Their work was not based on Finn et al. [28], but rather introduced a new matrix-based *soft-argmax* that outputs coordinates as values scaled between -1 and 1. color=blue]There are structural differences in how the two methods work, but I'm not sure if there are any differences in the effect they have on training. There must be, but if there are, I'm not sure if there is much written about it. color=orange, author=Vincent Brouwers (dev comment)]TODO: Write about how these two methods differ. While both approaches are not shown any heatmap examples, they cause networks to learn these implicitly Figure 2.5.

Sun et al. [87] later showed how the approach by Luvizon, Tabia, and Picard, which they dubbed *integral regression*, can be used to transform any heatmap-based method into a regression model. This effectively makes every heatmap model end-to-end trainable, removing the disparity between training and inference performance. It also eliminates the aforementioned quantization issue.



(a) Example image and with pose overlay

(b) Training target for heatmap matching

(c) Heatmap learned implicitly with DSNT

Figure 2.5: Demonstration of the implicitly learned heatmap with DSNT. Both heatmaps represent the location of the *neck* keypoint. Image (b) is a 2D Gaussian rendered at the ground truth location, whereas (c) is learned freely by a model. From *Numerical Coordinate Regression with Convolutional Neural Networks* [67]

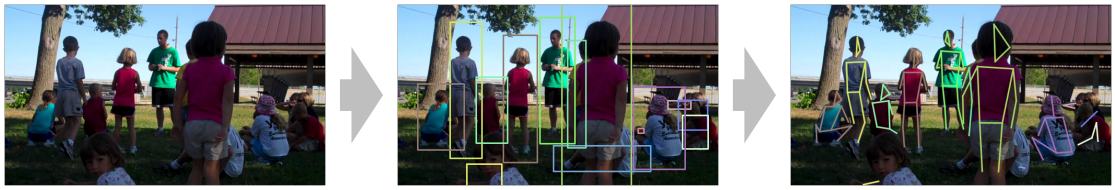
2.1.2 Multi-Person Pose Estimation

Multi-Person Pose Estimation requires not just the correct skeletal estimation for one subject in the picture, but it also has to deal with and distinguish many different subjects. This difference makes it so that one image can contain multiple instances of each joint, causing the normal single-instance models to fail. Two approaches are commonly used to address this problem:

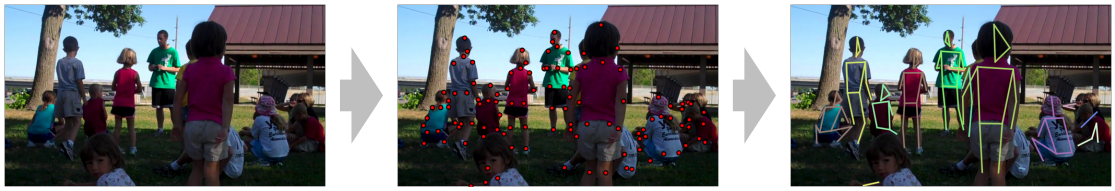
With the `top-down` approach , a person-detector provides bounding boxes for each visible human. The crops of these boxes are then fed into the keypoint-detector. This approach is conceptually the easiest and provides the highest accuracy. It does come with some drawbacks, one of them being that its performance is directly related to the number of people in a frame. It also heavily relies on the accuracy of the person-detector, which often fails with crowded images [43]. Overlapping bounding boxes can also result in skeletons where the joints belong to different people.

With the `bottom-up` approach , all joints are located in a single pass. A second algorithm is then used to group the joints into distinct skeletons. This approach is less accurate than the top-down approach but is considerably faster, even more so as the amount of subjects increases. It

does require complex joint-association algorithms. On the other hand, it also does not depend on the accuracy of the person-detector and often performs better on complex poses [43]. The joint-matching algorithms also handle crowded scenes better, though these models can also struggle to annotate very large and very small silhouettes [53] (relative to the frame).



(a) With a *Top-down* approach, individual subjects are identified and extracted, after which a single-person pose estimation model is used to predict the poses.



(b) With a *Bottom-up* approach, all visible joints are detected at once, which are then used to construct the most probable skeletons.

Figure 2.6: Difference in the localization process for multi-person scenes between top-down and bottom-up approaches. Image and annotations extracted from PoseTrack [4].

2.2 Design of a pose estimation model

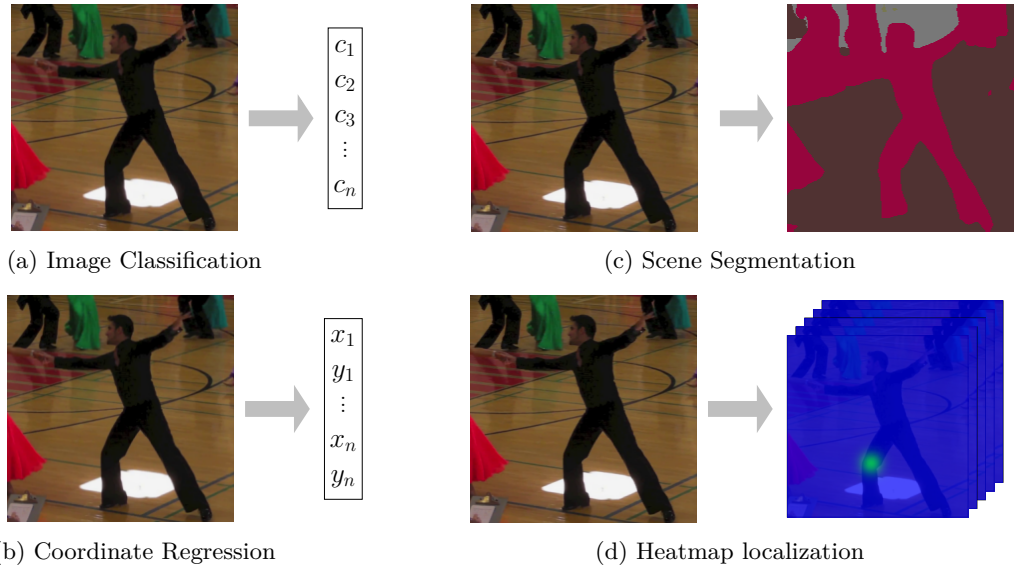
The goal of regression-based HPE methods has much in common with that of image-classification models; image-classification models aim to predict the numerical likelihoods that an image is part of n classes (Figure 2.7a), keypoint regression models predict the numerical locations of features in the image (Figure 2.7b). Heatmap-based HPE models, on the other hand, are more closely related to segmentation networks; both types of problems require the production of spatial maps that project an aspect of the original image (Figure 2.7c and Figure 2.7d).

The aforementioned similarities in objectives are also reflected as similarities in network architectures. Both classification and regression models rely on convolutional layers, often combined with pooling layers, to downsample and extract features from the images. Fully-connected layers then transform these features into the numerical output values. The keypoint-regression models implemented in DeepPose [89], for example, only differ from the AlexNet [50] classification model in the dimensions of their layers. Segmentation and heatmap models also rely on convolutional layers to process image information. Though since these models need to construct high-resolution output maps, downsampled features are often upsampled again in a later layer. Advances in either of the two fields often carry over to the other [54, 97, 92].

2.2.1 Design of a heatmap-based model

Convolutional neural networks work by convolving filters over a feature or input map.

This makes it so that features in resulting feature maps are only affected by a local area of features or pixels from the previous layer. This field of vision, or "receptive field" is stackable through multiple convolutional layers and determines how much context a feature has access to. In pose estimation, large receptive fields are crucial to capture long-range interactions between body parts [93]. This need for large receptive fields is shared among many computer vision problems that also want to base predictions on as much context as possible. That is, together with the computational benefits, why generally all CNNs use a deep stack of convolutions and pooling



(a) Image Classification

(c) Scene Segmentation

(b) Coordinate Regression

(d) Heatmap localization

Figure 2.7: Comparison of the input and output of keypoint localization methods to the input and output of similar techniques. Image classification models (a) aim to regress the likelihood of an image belonging to classes; coordinate regression models (b) aim to regress the location of keypoints in an image. (Semantic) scene segmentation models (c) aim to draw (semantic) boundary regions on an image; heatmap localization models (d) aim to draw Gaussian kernels on an image.

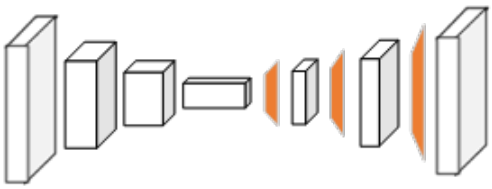
layers to compress the feature sizes [85, 33]. We want to condense as much information in these dense feature maps, which is why a halving in resolution is often paired with a doubling in feature depth [85, 33, 81]. This approach is excellent as-is for classification and regression tasks, heatmaps need to be of a sufficiently high resolution to minimize quantization errors. As such, these thus cannot be directly produced from downsampled features.

A common approach to create high-resolution heatmaps looks like an encoder-decoder model. In this context, Encoder-decoder models use the aforementioned downsampling method for feature extraction, before using an upsampling decoder to recover a high-resolution heatmap. In these models, the encoder can sometimes be any off-the-shelf CNN architecture such as VGG [85] or a ResNet model [33], often already pretrained on another image perception task such as ImageNet [20] classification [94, 73, 11, 17, 74, 75].

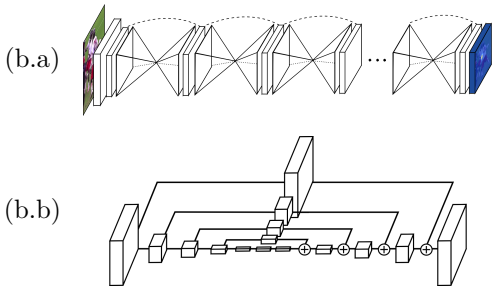
A simple, yet surprisingly performant, example of such a model is the aptly named "Simple-Baseline" model by Xiao, Wu, and Wei [94]. Visualized in Figure 2.8a, this encoder-decoder-style model uses an unmodified ResNet-152 [33] encoder which is pre-trained to classify ImageNet photos. The coupled decoder consists of just three transposed convolutional* layers for upscaling and a single 1×1 convolutional layer to create the final heatmap. These transposed convolutional layers in essence perform the opposite operation to normal convolution layers; a standard convolutional layer multiplies filters with a region of features and extracts a single feature, transposed convolutions multiply individual features with filters and extract a region of features.† When combined with striding, this results in a learnable upsampling operation, as is the case in SimpleBaseline. Each transposed convolutional layer in this model has 256 4×4 filters with a stride of 2×2. The final 1×1 convolution has the same amount of filters as the number of joints and outputs the final prediction heatmaps. Following Newell, Yang, and Deng [66] and Chen et al. [17], images

*Transposed convolutional layers are also often, but mathematically incorrectly, referred to as "deconvolutional" layers [24, p 20]. In practice, "deconvolutional" and "transposed convolutional" are often used interchangeably.

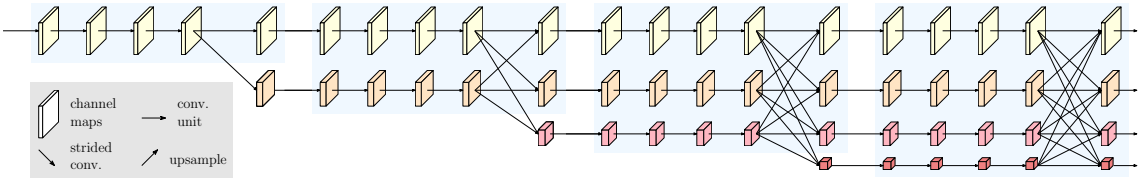
†The produced regions overlap when the filter size is larger than the stride, in which case the final value is the sum of overlapping regions. The filter size should be divisible by the stride, as else an "uneven overlap" results in checkerboard-like patterns in the output features [70]



(a) Schematic representation of SimpleBaseline [94]. This simple, yet effecive encoder-decoder model consists of a standard ResNet encoder, followed by several transposed convolutional layers that perform a trainable upscaling operation. This stands in contrast to the earlier Stacked Hourglass [66] (Figure 2.8b) and Cascaded Pyramid Network (CPN) [17] architectures, which nearest-neighbor and bilinear upsampling, respectively, instead. Image adapted from [94].



(b) The Stacked Hourglass network (b.a) network consists of eight consecutive U-Net-like hourglass modules (b.b) that condense the input before reconstructing a matching heatmap. The condensed features are upsampled and combined with more detailed information from earlier layers via skip-connections. The stacking of the modules enables repeated refinements of the joint localizations. The modules are trained with intermediate supervision, where the heatmap reconstruction loss is not just computed at the final module, but also for each intermediate module. Images from [66]



(c) Schematic representation of HRNet. Contrasting to the (repeated) downsampling and upsampling of (a) and (b), HRNet instead maintains a high-resolution representation throughout the network. It further maintains separate subnetworks of varying lower resolutions that exchange information at set phases in the network.* HRNet derivatives currently make up of three of the five top-performing models on the COCO Keypoints Leaderboard [19]. Image from [92].

Figure 2.8: Visualizations of three heatmap-based CNN HPE models.

are evaluated in both the in their original form and flipped horizontally to improve estimation accuracy. For each keypoint, the location of the highest confidence prediction is taken and offset a quarter towards the lower-confidence prediction.

Though many heatmap-based HPE models follow the practice of downsampling and then recovering the high-resolution heatmaps from the encoded low-resolution feature, this is not followed by all. HRNet [92, 86] instead maintains a high-resolution representation throughout the network and is currently the base model for three out of five of the five top-performing models on the COCO Keypoints Leaderboard [19]. Instead of downsampling and upsampling the image data, HRNet maintains separate subnetworks of varying resolutions (see Figure 2.8c)that exchange information at set phases in the network.* As in most networks, high-resolution feature maps are spatially precise, whereas low-resolution maps are semantically strong. This repeated "multi-resolution fusion" boosts both qualities across all the resolution levels.

*The authors refer to this exchange of information as "multi-resolution fusion"

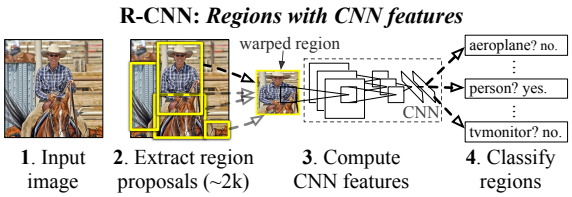
2.2.2 Design of a multi-person pose estimation algorithm

Single-person pose estimators can rely on just their localization networks. Both top-down and bottom-up multi-person estimation approaches, however, require additional algorithms to discern the different subjects in frame. Top-down methods require some form of bounding-box detector for persons, whereas bottom-up approaches use various different heuristics for keypoint matching.

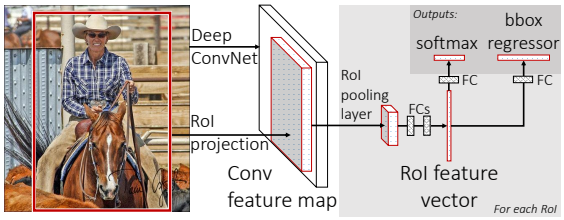
Top-Down

HPE methods are very reliant on accurate people detections. Undetected people will not be annotated and erroneous detections will result in false joint detections. The detections are directly processed by the subsequent single-person pose estimator, meaning there is no recovering from an incorrect detection. The manner in how these detections are performed does not matter for the rest of the pipeline, however.

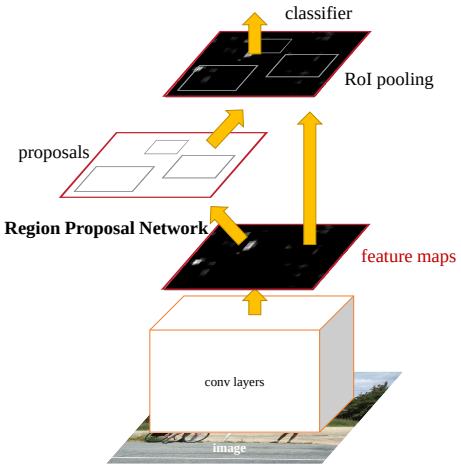
In general, any single-person pose estimation model can be combined with a human detector to add multi-person pose estimation capabilities. As an example: the SimpleBaseline model from the previous section was combined high-performing human detector to achieve SOTA performance on the multi-person COCO [19] dataset. Following G-RMI [72] and later followed by HRNet, the multi-person version of SimpleBaseline employs a (pretrained) Faster R-CNN [79] object detector that identifies bounding boxes around human subjects in each image. The image data inside each of these bounding boxes is then cropped out and processed via the normal single-person SimpleBaseline model. The fully-convolutional SimpleBaseline model can process images of varying dimensions, though some architectures, often regression methods such as DeepPose, require an intermediate transformation that warps the detected area to a processable form. In the end, the estimated coordinates, which are local and relative to the cropped area, are merged with the location of the detection to retrieve the final global coordinates.



(a) Original R-CNN architecture. Image from Girshick et al. [31].



(b) Fast R-CNN architecture. Image from Girshick [30].



(c) Faster R-CNN architecture. Image from Ren et al. [79].

Figure 2.9: Variations of the R-CNN architecture.

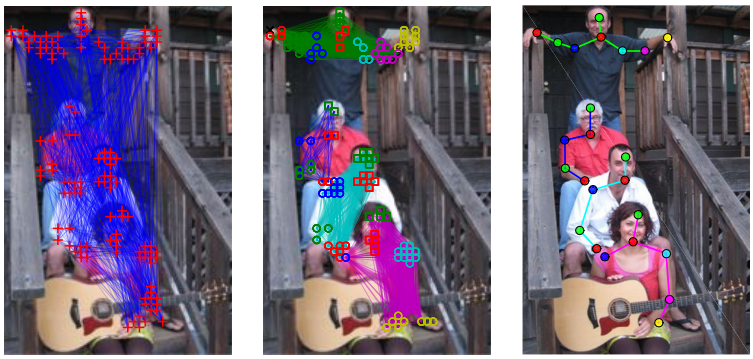
Top-down pose estimation can be performed with any off-the-shelf human detector, though the "Faster R-CNN" model by Ren et al. [79] is a commonly used [43, 53, 86] example. A Faster-R-CNN model is, as the name implies, a faster version of an R-CNN model. Visualized in Figure 2.9a, an R-CNN consists of two parts: a Region-Proposal Model (RPM) and a CNN classification model [31]. The classification model attempts to classify the contents of each region, after which sufficiently identifiable regions are kept as the final detections. The original R-CNN model (Figure 2.9a) used an RPM algorithm called selective search to propose regions and ran an AlexNet [50] classifier over each region individually. As AlexNet can only process images of a set resolution, the detection regions were first warped onto a square 227×227 format. A later improvement in the form of

Fast R-CNN [30] (Figure 2.9c) required just one feature extraction step to be performed by the classification model, now VGG-16 [85]. A special Region of Interest (RoI) pooling method after the feature extraction process allowed individual regions to be pooled into a set-size feature map that could be classified as if the images were warped before the feature extraction. Faster R-CNN is the latest progression and introduces a CNN-based RPMthat can be combined with the classification model into one such that they share the majority of their parameters. The classifier itself is also upgraded again, this time being ResNet-101 [33].

Bottom-Up

DeepCut [75] is an early and influential bottom-up pose estimation algorithm. Most multi-person methods at the time of publishing were top-down approaches whose performance suffered accuracy losses on people in close proximity. Overlapping detection boxes can cause the subsequent estimation models to predict skeletons with joints belonging to different persons. Pishchulin et al. therefore introduced DeepCut, a bottom-up approach that does not share this problem. Instead, it localizes all joints globally and uses an Integer Linear Program optimization process to connect the joints in the most likely manner.

Deepcut uses a modified R-CNN to generate body part candidates. These candidates are then fed into a CNN with multiple outputs that acts as both a regression and classification model. This predicts likelihoods for each of the body part classes C and provides more accurate relative coordinates in a manner similar to DeepPose [89]. All predictions are transformed into a fully-connected graph (see Figure 2.10a) and processed using an ILP solving algorithm. This algorithm attempts to find the most likely subgraphs such that each subgraph represents exactly one skeleton (see Figure 2.10c). It does this by finding the cheapest solution such that all connected keypoint nodes satisfy the following constraints: 1) Each keypoint can be of at most one joint type, classless keypoints are suppressed, 2) suppressed keypoints cannot be part of a body, 3) a body can have at most one of each joint type, and 4) if keypoint k and k' are part of the same body, as are k' and k'', then so are k and k''. As more solutions are possible, DeepCut attempts to find the cheapest solve where the cost of keypoints is defined by the likelihood that joints are of the predicted class and the cost of edges is defined by the likelihood that one joint solve implicates the other.



(a) (b) (c)

Figure 2.10: Visualization of the DeepCut algorithm. (a) shows the fully-connected graph of joint, which are clustered in (b) s.t. each subgraph contains proposals for just one subject. Color and shape combinations indicate different joint types. (c) depicts the final solved skeletal predictions. Note that Deepcut does perform an explicit subject clustering stage as depicted in (b), as this is performed through the ILP solving. Images from [75].

It was the first CNN bottom-up approach and its robustness awarded it SOTA on both the COCO and MPII HPE datasets. The ILP solving, however, is an NP-hard problem that can take tens of hours per image to solve. A later revision by the same authors called DeeperCut [41] provided much-improved performance by substituting the global ILP solving by an incremental multi-stage variant. This 3-stage solver first just processes joint proposals for head and shoulder keypoints. When this problem is solved, elbows and wrists are added, followed by hips, knees, and

ankles. This splitting significantly reduces the number of edges in the fully-connected graphs and allows for the ILP solving to be completed in around 270 s/frame*, three orders of magnitude less than the original DeepCut.

OpenPose , a popular open-source HPE framework by Cao et al. [10], uses a significantly faster method for bottom-up multi-person pose estimations. Based on earlier work of the same authors, CMU-Pose [9], OpenPose does not rely on the same computationally heavy ILP solving as deep-cut to group keypoints. Instead, it relies on Part-Affinity Fields (PAFs) to match related joints. These PAF maps are 2D vector fields that indicate both the direction and location of limbs. Just as with the heatmap-predictions, the PAF maps are generated by CNNs. Where the heatmap predictions localize individual keypoint types, the PAF maps are used to show a directed relatedness between the joints of a limb.

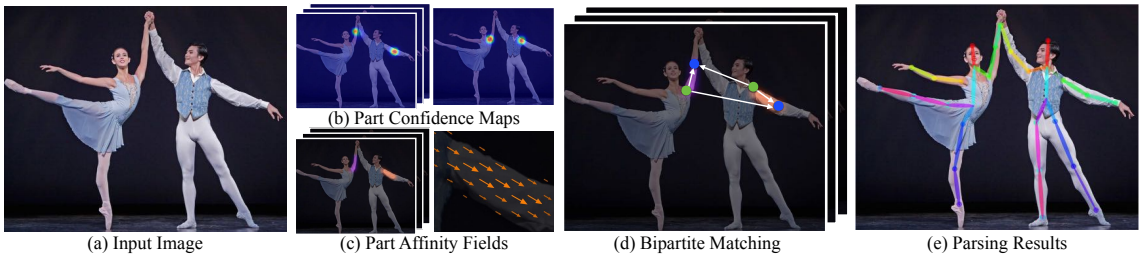
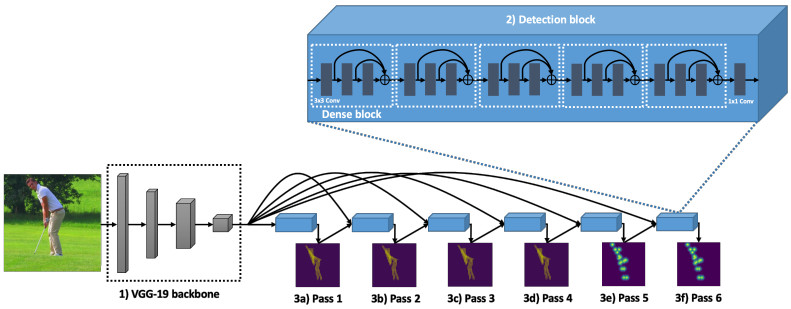


Figure 2.11: Visualization of the OpenPose (and also CMU-Pose) pipeline. The input image (a) is converted into localization heatmaps (b) and part affinity fields (c) via CNNs. The PAFs are then used to match related keypoints belonging to the same limbs (d), finally resulting in the skeletal representations from (e).

Both the localization heatmaps (which the authors refer to as “Part Confidence Maps” for a similar termonology to “Part-Affinity Fields”) and the PAFs are produced by an iterative refinements with intermediate supervision. Inspired by Wei et al. [93], and similar to the Stacked Hourglass [66] model from Figure 2.8b, OpenPose relies on multiple stacked CNN stages that iteratively predict and refine the target maps (see Figure 2.12a). Contrasting to these earlier methods, however, OpenPose does not produce part confidence maps at each stage. Instead, the first four stages solely work to predict the PAFs (Figure 2.12b), whose result is then used to predict the part heatmaps via the next two stages. CMU-Pose, the predecessor of OpenPose, predicted both map types at each inference stage simultaneously, for which it effectively used two separate CNNs. The authors however found that the PAFs did not require as many refinement steps, as well as that the part confidence maps could be predicted from the PAFs. color=blue]Should I use numbers or words to describe amounts like the number of stages? I know that spelling ”four” is more formal than writing ”4”, but I also know that technical texts often prefer numerical notation...

Finding an optimal skeletal parse in a graph of joints (such as in Figure 2.12c.2) in NP-hard [10]. Combined with the PAFs, OpenPose instead uses a greedy parsing algorithm that approximates the global solution at a fraction of the cost. It firstly creates a bipartite subgraph (Figure 2.12c.4) for each pair of joint types that can be connected. The edges in these graphs are then weighed by sampling the corresponding Part-Affinity Field along the line that each edge draws on the image. This affinity score directly corresponds to the likelihood that the edge represents a limb connection. The greedy picking algorithm then iteratively picks the highest-scoring edge that does not share any keypoints with earlier picked edges. Finally, all picked edges and nodes are merged into the full skeletal representations as visualized in Figure 2.11.

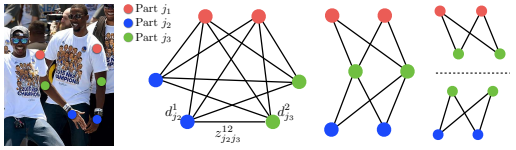
*Average duration taken over all images in the MPII test set



(a) Diagram of the OpenPose architecture. The first four stages extract the PAF maps, followed by two stages that extract the confidence maps. Each stage is supervised with their respective ground-truth map type and consists of five blocks of 3×3 convolutional layers that have their outputs concatenated. The input image is first processed by a feature extractor based on the first 10 layers of VGG-19. Note that this limits the resolution of the OpenPose predictions, which is a prominent example of the quantization issue for heatmap-based models presented in Section 2.1.1. Image from Groos, Ramampiaro, and Ihlen [32].

Stage 1 Stage 2 Stage 3

(b) Accuracy improvements throughout multiple stages of PAF refinements in OpenPose.



(c.1) (c.2) (c.3) (c.4)

(c) Creation of biparte graphs (c.4), compared to

Figure 2.12: Images from Cao et al. [10].

2.3 Pose Datasets

For a long time, progress in the field of computer vision has gone hand-in-hand with the development of new datasets. Ever since the release of ImageNet in 2009 [20], the field of computer vision has experienced a boom in both performance and attention. Where previous datasets had no more than tens of thousands of images [49, 52], ImageNet had 3.2 million images. This amount of data enabled a boom into data-hungry CNN research, which is still ongoing.

While the ImageNet dataset enabled many developments to happen for CNNs, the data was mainly useful for the classification of objects. Datasets are inherently costly to create, as manual annotation is a labor-intensive process. Where ImageNet was expensive to create due to its size, the images themselves were mostly easy to manually classify. Pose estimation datasets require more effort to annotate each sample, as annotators need to place each keypoints on the right location for each image. This results in even the largest HPE datasets being orders of magnitude smaller than similarly popular image classification datasets.

The following two sections discuss the most widely HPE datasets. We emphasize the methods used to collect them and the possible biases that training on them could introduce against children.

2.3.1 Image datasets

Leeds Sports Poses (LSP) [44] is one of the earlier datasets that aimed to provide a dataset that is realistically sized to train models varied human poses. They collected their videos by querying Flickr for images with one of seven sport tags and scaled these such that the annotated person was around 150 pixels tall. Though nothing constrained the subjects to be adults, the resulting 2000 images contain almost no child subjects. The LSP-Extended [45] dataset was created in a similar fashion by the same authors, after realizing non-upright poses, like those found in gymnastics and parkour, were especially challenging.

Frames Labeled In Cinema (FLIC) [82] is another early effort at creating a large pose dataset of high-resolution images. The authors note that earlier datasets such as H3D [6] and PASCAL VOC [25] contain mostly images of insufficient resolution. Their new FLIC dataset contains 5003 high-resolution images, of which 3987 are used for training. The images are all extracted are frames from 30 different Hollywood movies. As many of the following datasets, the individual images were annotated via Amazon Mechanical Turk (AMT). Due to the choice of movies, this dataset does not contain any images of child subjects. It is also annotated on just 10 upper-body joints and is manually filtered to exclude occluded or non-frontal subjects.

An unfiltered version, FLIC-Full, was also presented in the same paper. However, this set contains images both identical and similar to those in the FLIC test set. The authors of FLIC-Plus [88] excluded all frames from FLIC-Full that originate from the same scenes as those in the FLIC test set and unioned this with the original FLIC train set.

MPII Human Pose (MPII) [3] is a one of the SOTA HPE benchmark datasets [18]. It contains manually selected frames of videos queried from YouTube and annotations were generated via AMT. The YouTube queries were derived from an activity compendium built from PA patterns of adults [2], introducing an early bias towards them. The dataset contains 24,920 images and a combined 40,522 poses. It also provides the previous and following frames for models to facilitate the use of motion information.

Common Objects in Context (Microsoft COCO) [55] is a large and varied dataset with annotations for object detection, scene segmentation, dense pose* detection and keypoint detection. The 2014 paper introducing COCO describes how the images were gathered by searching in image repositories such as Flickr for the objects the authors originally wanted to detect. At this time, the keypoint dataset was only speculative and there is thus not much information about it in the paper. The collection process does not indicate a fundamental bias against children. Though of course largely because the paper focuses on object recognition, there is also no mention that care was taken to ensure a proper demographical representation.

2.3.2 Video datasets

VGG Pose datasets are a family of pose datasets constructed by the Visual Geometry Group from Oxford University. Due to our focus on monocular RGB datasets, we exclude ChaLearn and focus instead on BBC Pose [14], Extended BBC Pose [74], Short BBC Pose [13], and lastly Youtube Pose [15].

BBC Pose contains footage of British Sign Language (BSL) interpreters of the British Broadcasting Corporation (BBC). Its has semi-automatically annotated training videos, with manually annotated test and validation videos. Short BBC Pose is similar but contains just five training videos. Extended BBC Pose, however, adds 8× more, albeit slightly noisier, training data with fully-automatically generated annotations. Youtube Pose, consists entirely of manually annotated YouTube videos on a range of activities. Though the BBC Pose variants are large and YouTube Pose is varied, the subjects are all adults. They also all focus on just nine upper-body joints.

Joints for HMDB (J-HMDB) [42] is a joint-annotated subset of the HMDB Action dataset [51]. The original action dataset contains 51 categories of videos from various online sources. J-HMDB adds joint annotations for videos of 21 categories, limited to videos where the actors are prominently visible. As with previous datasets, children are sparsely represented in J-HMDB. Part of this can be explained by the inclusion of actions such as “shoot gun” and “pull-up”, generally not child activities. With the exception of the surprisingly infant-dense “push” category, most other actions, however, such as “clap” and “walk” also contain few children.

*Dense human pose estimation aims at mapping all human pixels of an RGB image to the 3D surface of the human body. From *DensePose* [21].

Penn Action [99] is an action dataset that provides meta-annotations such as bounding boxes and pose keypoints. The authors do not go in-depth into the way the videos were selected, apart from that they originate from online sources. It is thus not possible to identify a bias in this process. The resulting clips, however, contain little to no children.

PoseTrack [4] is HPE benchmark with an additional focus on multi-person articulated tracking. It expands upon MPII by including five seconds of footage around the original MPII frames, selecting crowded and active scenes. Besides pose, this dataset also tracks individuals throughout the clips. It also provides unique ignore regions for crowds too complex to annotate, which can be used to exclude *false positives* during training and testing. As it is based on MPII, any early biases present in that dataset, also carry over to PoseTrack.

color=orange, author=Vincent Brouwers (dev comment)] TODO: Add Human3.6M as mainly an evaluation dataset.

Type	Dataset	Train Images		Total Images		Joints	Source
Images	FLIC [82]			3987		5003	10	Movies
	FLIC-Full [82]			20,928		20,928	10	Movies
	FLIC-Plus [88]			17,380		17,380	10	Movies
	LSP [44]			1000		2000	14	Flickr
	LSP-Extended [45]			10,000		10,000	14	Flickr
	MPII [3]			24,920		24,920	16	YouTube
	COCO [55]			118,287		123,287*	17	Flickr
	Sciortino et al. [83]			1176		1176	22	Video portals
		Train		Total			
		Sequences	Images	Sequences	Images		
Video	BBC Pose [14]	10	610,115	20	612,115	9	BSL interpreters
	Short BBC Pose [13]	5		15		9	BSL interpreters
	Extended BBC Pose [74]	85	5,782,140	92	5,784,140	9	BSL interpreters
	JHMDB [42]	$\sim$ 660†	31,838	928	31,838	15	Various sources
	Penn Action [99]	2326	163,841	2326	163,841	13	Video portals
	YouTube Pose [15]	50	5000	50	5000	7	YouTube
	PoseTrack [4]	593	43,603	1138	109,513	15	YouTube

Table 2.1: Qualitative comparison of currently available HPE datasets. Some datasets contain samples meant specifically for testing or validation. These are excluded in the "Train" columns, but included in the "Total" columns. For video datasets, the sequence count indicates the amount of clips. These may be clips from the same source video.

All mentioned datasets were collected because the authors saw a need for a larger or more varied dataset. In their methods, however, little consideration was put into having the data reflect the demographical makeup of our population. Many of these datasets are built by querying activities mainly performed by (or at least recorded of) adult subjects. color=orange, author=Vincent Brouwers (dev comment)]Think of how to finish this

2.4 Child pose estimation

Though most HPE studies and datasets are focused on adult subjects, there are some with a focus on children. To our knowledge, there are is at this time just one earlier work in this area that created a child-centric RGB monocular pose dataset. In 2017, Sciortino et al. introduce a

*COCO has over 200.000 images annotated with keypoints, but not all are publicly available. The COCO test set is withheld due to the ongoing competition.

†J-HMDB has multiple train/test splits: 660/268, 658/270 and 663/265

benchmark dataset of child and infant subjects. They use this to show that HPE models trained on adult-biased datasets, perform measurably worse on this domain. Their dataset is partly comprised of videos from an action recognition dataset focused on early autism detection in children, *Self-Stimulatory Behaviour Dataset (SSBD)* [77]. The videos of this dataset came from various video portals, including YouTube. Sciortino et al. disregarded videos from SSBD where people were interacting or had strongly truncated poses. They further padded their dataset with videos by manually querying YouTube with keywords for certain expressive activities and variations of "child" or "toddler".

The final dataset contains 1176 images from 150 unique videos with 104 unique subjects. Retraining these models on this dataset is difficult due to its limited size (Table 2.1) and is thus not something the authors attempted. The authors concluded there to be an accuracy drop across several different models when tested on their child dataset. They, however, did not address that any difference in performance could also be the result of different strategies for collecting and/or annotating their data.

We requested access to the dataset but received no reaction from the first, nor from the second author. Further lack of reviews thus limits our capabilities to discuss this dataset.

Chapter 3

Data and Methods

To answer the research questions defined in Section 1.4, we need a sufficiently large kid-specific dataset that can be used in training deep-learning models. This section describes the manner in which we collect our data and the reasoning behind our decisions. We call this new dataset *Kinetikids-pose*, sharing its name after the child action dataset by Olalere [71] with which part of it is derived.

Kinetikids-pose contains 1064 images of 1384 different pose-annotated children within the ages 0-12 (pre-pubescent). In contrast to many other child-specific datasets, we choose to share this dataset publicly. [†] The joints are all annotated in the COCO keypoint format to facilitate straightforward usage with existing models developed for the COCO dataset.

The dataset is composed of two separate sources. The first part of the dataset, 219 images, consists of frames extracted from 219 different *Kinetikids* videos. These are YouTube videos containing children performing one of 38 selected sporting activities. The keypoint annotations for this section are crowdsourced via the Amazon Mechanical Turk crowdsourcing platform. The remaining 845 images are gathered by querying Google Images. We used the same 38 sporting activities to generate the queries, though no effort was made to ensure these activities indeed occur in the final images. The keypoint annotations for this section are labeled by students and staff of Utrecht University.

3.1 YouTube Videos

The first part of *Kinetikids-pose* consists of frames extracted from videos of the *Kinetikids* dataset. This section describes the process from the collection of the videos and up to the moment we have pose-annotated frames, also visualized in Figure 3.1.

3.1.1 Video Collection

We first compile a list of sports activities to include in this dataset. We picked out 38 sport activity labels from the defined sports categories in Kinetics-400. We choose the sports category based on our hypothesis that there should be an observable difference between how a sporting activity is performed by an adult as opposed to a kid. We work with only one category in this project because of time and resources constraint.

After compiling the activity labels for Kinetic-kids, we define specific query lists tailored to search for videos with kids performing these activities (see Appendix A). We tailor our query to target the age group we are interested in e.g *basketball game in pre-school* or *kids dunking basketball*. These queries are then searched for on YouTube. Before downloading the returned videos, we check that the video is at most 100MB, This is to filter out professionally shot and heavily edited videos. The non-professionally shot videos are less edited and are more depicting of the real world. Furthermore, we only download videos that had a resolution of 480p or 720p.

[†] `https://drive.google.com/drive/folders/1nQooLjW1c8bWfAz2ilj2ojJl8B8Db_cN`

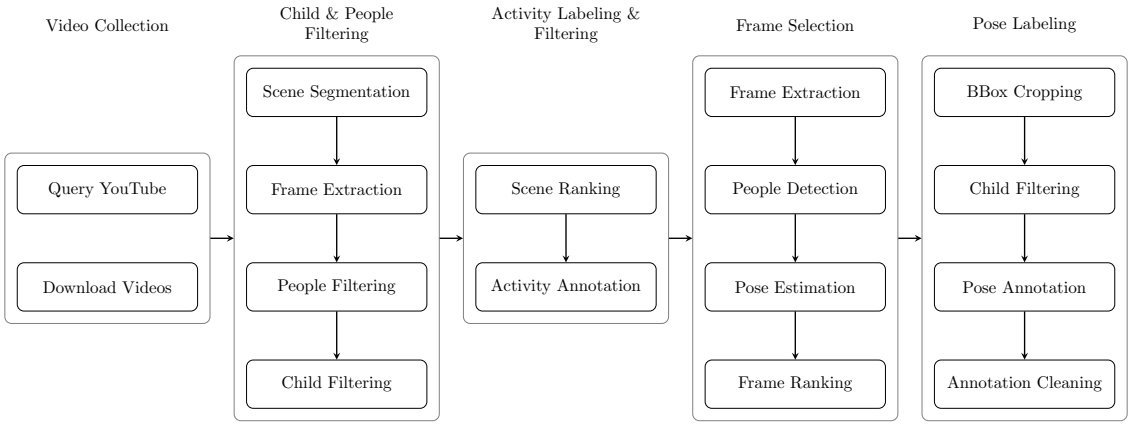


Figure 3.1: Flow diagram of all processing steps made gathering the YouTube section of *Kinetikids-pose*

We selected videos with these resolutions so they could also be useful for collating pose estimation datasets. The videos that meet both criteria are downloaded and saved for pre-processing and annotation.

3.1.2 Child & People filtering

Once we have downloaded the videos, we firstly perform a scene detection step on each video. oi1Using *PySceneDetect* [12], we check for differences in the HSV space between consecutive frames. If the difference in average HSV pixel values between consecutive frames is greater than 30%, we presume a scene change. We start by splitting the videos into scenes as we deem that actions that appear across the scene boundaries are not informative for us.

After performing scene detection for each video, we select three evenly spaced frames per scene that is longer than one second and pass these through a pre-trained YOLO-V3 [78] model for people detection. This detection model achieves a person detection mAP of 50.3%. Its accuracy is less than the 56.4% mAP that the Faster R-CNN model employed by Simple Baseline and HRNet achieves, but requires only a fraction of the computational cost. We take this step to eliminate scenes that do not have people in them, as as we are only concerned with scenes containing human activities. Detections with a confidence score of less than 70% are discarded.

Child Filtering

Since our goal is to end up with a kid-specific dataset, we filter out scenes that are unlikely to contain children. Most current age recognition techniques are limited to faces (via datasets such as MORPH [80], CACD [16] and FG-NET [29], AFAD [69], and UTKFace [100]) or to voice recognition (*TODO: I have one or two papers about this in my library, but I am not familiar enough with it to say "These papers are good examples!"*). For our purpose, however, we cannot rely on facial features, as this would induce an obvious bias towards front-facing subjects, nor can we rely on voice features as possible speech in the video does not have to originate from our subject. Work exists that distinguishes children from adults on anatomical differences by use of their different head-to-body ratios [39, 40], though these models are not public and the authors ignored our requests for access. To our knowledge, there is no other work published to specifically identify children based on full-body queues, that works from various angles, and requires no specialized hardware (such as 3D cameras [5]).

Traditional image recognition models are often trained in a supervised manner. These thus require a sufficiently large (hand-crafted) labeled dataset to predict the probability that an image belongs to one or more class labels. We instead employ the recently developed zero-shot model CLIP [76], by OpenAI, to detect children. CLIP is trained on 400 million automatically-collected image-text pairs and outputs cosine similarities between pairings. By comparing the similarities of an image to several hand-crafted indicator sentences, we can map an image to the class it is

most similar to. We manually label a subset of our data and finetune our methods of data input to optimize CLIP's ability to predict if a bounding box contains a child or adult. This test set contains 1350 people, 1001 adults, and 349 children.

As a baseline, we developed basic prompts describing our images, as was done in the CLIP's study. We take a crop of our images for each bounding box predicted by the people detection step, with a margin of 20% in both the width and the height. Our images are crops taken from‘ the bounding boxes detected by the YoloV3 detector mentioned in the previous section. The prompts include sentences formed like *a photo of a <label>*, where label was one of infant, toddler, child, teen, or adult. As CLIP accepts only 224×224 images, we rescale the crops such that their longest side fits these restrictions size and pad the edges to make them square. As a baseline padding strategy, we mirror the image along the edges of the image (reflect padding).

Using the same margin scale and padding method as the baseline, we can optimize the way we formulate our indicator sentences. As a first step, we use the same prompt templates as used in the CLIP study for ImageNet. Following their methods, we formulate multiple sentence variations (such as *a bad photo of a <label>*, *a photo of a large <label>*) per label and average the CLIP text embeddings per label. This gave an increase in CLIP's performance over when we only use the baseline prompts. Finally, we append the words *doing <sports>* to all the ImageNet prompts (ImageNet+sports); here *<sports>* is replaced with one of our sporting categories. This means the prompts (text) in the image-text passed into CLIP looked like *An image of a child doing badminton*.

Next, we vary the margin scale between 0% and 30% in increments of 10% and settled on the initial value of 20%. Lastly, we vary the padding function as a further optimization strategy. In addition to *reflection padding*, we tried: *zero-padding*, where the color values of the padded edges of the images are set to 0 (black); *one-padding*, where the padded edges' color values are set to 1 (white); and *reflection padding*, where we repeat the last line of pixels of an image's original edge until the padded space is filled.

Based on the result of the ablation study, our final model configuration uses ImageNet+sports prompts, a margin scale of 0.2×, and zero-padding (See results in Table 3.1).

	Prompt type	Margin scale	Padding	AP	AUC
	"a photo of <label>"	20%	Reflect	0.689	0.421
	ImageNet	20%	Reflect	0.712	<u>0.443</u>
	ImageNet + sports	**20%**	**Reflect**	<u>0.769</u>	0.434
	ImageNet + sports	**0%**	Reflect	0.750	<u>0.460</u>
	ImageNet + sports	**10%**	Reflect	<u>0.765</u>	0.453
	ImageNet + sports	**30%**	Reflect	<u>0.765</u>	0.411
*	ImageNet + sports	20%	**Zero-Padded**	0.813	<u>0.468</u>
	ImageNet + sports	20%	**Replication**	0.804	0.441
	ImageNet + sports	20%	**One-Padded**	<u>0.814</u>	0.458

Table 3.1: Abblation study of hyperparameters for our CLIP child detector. Per section, **bold** text indicates the variable of interest, <u>underlined</u> results indicate the best scoring results per section. The row indicated with a * shows our final configuration

To finetune the predictions of the model such that it differentiates between prepubescent children and teens, we also use an ensemble of indicator labels per class. The labels "infant", "toddler", "child" all indicate our desired "child" class, whereas "adult" is indicated by "adult" and "teen". Instead of cosine similarities per label, we want to have the model output a single value in the range of $[0, 1]$ as our child probability. Formalized in Equation (3.1), we calculate this by taking the cosine similarities Z and pick the maximum cosine similarities $\mathcal{Z}$ of the labels for both our "child" and "adult" classes. We convert these into probabilities via a softmax step σ. Finally, since this is a 2-class problem, it suffices to just use the probability of our "child" class.

$$\begin{aligned} Z &= CLIP(x,\ \text{labels},\ \text{prompt templates}) \\ \mathcal{Z}_0 &= \max(\{z \in Z \mid z \text{ is child label}\}) \\ \mathcal{Z}_1 &= \max(\{z \in Z \mid z \text{ is adult label}\}) \\ P &= \sigma(\mathcal{Z}) \\ P_{child} &= P_0 \end{aligned} \tag{3.1}$$

The goal of the child detector is to filter out the clear adults, without filtering out too many children. We choose to filter out all people for which the child detector scores a child probability of less than 40%. At this threshold, the estimator has a precision and recall of 66% and 79%, respectively, for the child class of our development set. Our development data contains three adults for every one child before child filtering. Assuming the distribution of children and adults from this subset is representative for our greater set of frames, we filter out 94% of the adult people.

	Children	Adults
Precision	66%	94%
Recall	79%	90%

Table 3.2: Precision/Recall table for the child detector at the 40% threshold

3.1.3 Activity Labeling & Filtering

To create a robust dataset of expressive poses, we will only annotate poses from scenes where a child subject is performing an action. The specifics of this process are further expanded upon in Olalere [71].

After the preprocessing steps from the previous sections, we remove all scenes that do not contain at least one frame containing at least one child. A ResNet-50 SlowFast model[26], trained on Kinetics-400[46] is then used to predict which actions are performed per remaining scene. A ranking algorithm then assigns each scene per video a score based on how dissimilar the predicted actions were to the expected actions that the video was queried for. After that, a greedy algorithm generates a sequence of consecutive scenes with the highest prediction dissimilarity, limiting the sequence length to 60 seconds. Lastly, this list of scenes is then concatenated into a new video clip.

Each produced video clip is presented to two distinct AMT workers together with five suggested action labels. The workers label the first clear instance that one of those actions occurs in the video, together with the timestamp. When the workers submit differing timestamps or action labels, the clip is presented to a third annotator.

3.1.4 Frame Selection

From the clips in *Kinetikids*, we select ten evenly spaced frames from the two seconds following the annotated action timestamp as candidate frames to annotate for poses. Similar to the method described in the previous section, how we apply a filtering step to ensure the poses present some difficulty to current SOTA HPE models. Instead of looking at the confusion of a single model, here we instead look at the disagreement between two different models.

The selected frames are processed by the same YoloV3 model as used in Section 3.1.2. We keep all people with a bounding box larger than 150 pixels tall and analyze them with two human pose detectors: SimpleBaseline and HRNet. We opted for top-down models so we can directly compare pose annotations per bounding box instead of having to match poses between annotations. We use the SimpleBaseline with the ResNet50 backbone with an image size of 256×192 and HRNet W48 with an image size of 384×288. These are the simples version and most complex version of their respective architectures.

We compare the pose estimations using the Object Keypoint Score (OKS) score to calculate the difference between the annotations. As the OKS is a non-symmetric measurement, we take the measurement both ways. To remove unchallenging poses, we only take bounding boxes where

the maximum of our two OKS scores is < 0.90. We also discard annotations where the minimum of our two OKS scores is < 0.10, as such a large disagreement mostly occurs blurry images or bad person detections.

Before the OKS filtering, we have 4179 frames containing 6676 people from 505 videos. After filtering, 977 frames with 1113 people from 323 videos remain. We group the frames per video and take the frame with the lowest average of the OKS scores. The remaining 437 detections from 252 videos are manually filtered to contain solely child images to arrive at a final 322 people from 221 videos.

3.1.5 Pose Labeling

We annotate the poses via the crowdsourcing platform AMT. AMT workers are presented with an image of a child and asked to place markers on each of the COCO keypoints (see Appendix B.1 for a full list). Per annotated pose, workers are paid \$0.07. With this rate, a fast annotator can earn an hourly wage of \$6.30 to \$8.40 when annotating one pose per 45 or 30 seconds, respectively.

We use a modified version of the Amazon Mechanical Turk default keypoint-annotation interface for this.* The default interface does not allow for associating keypoints to separate child instances, making it unsuitable for annotating multiple persons at once. Instead, we present workers with cropped images for each of the selected bounding boxes ask the workers to annotate only child the "child of interest". More specifically, we crop the image to an area 2.5× the scale of the bounding box and indicate the child of interest with a red outline 1.5× the scale of the bounding box. Shown in Figure 3.2, we found this to give a good trade-off between showing context and keeping the focus on the desired subject.

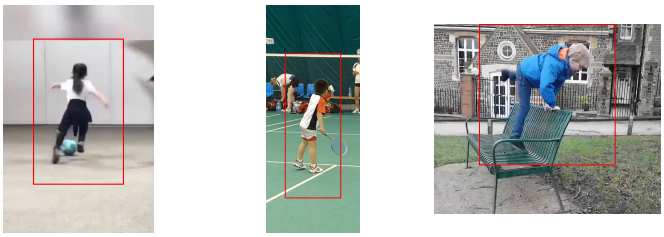


Figure 3.2: Sample of images shown to MTurk workers

Mechanical Turk Data Cleaning

The Amazon Mechanical Turk annotations are exceptionally noisy annotations. Besides obvious rubbish annotations, there are also plenty of other, commonly made mistakes.

Similar to Johnson and Everingham [45], we identify four commonly made labelling errors: **full-body left/right switched**, **face left/right switched**, **torso left/right switched**, **legs left/right switched**. Instead of solving these errors via an iterative bag-of-poses learning method, we state this as a simple optimization problem where we apply flipping transformations to minimize the difference between the annotations.

The difference between two annotations is defined as the product of modified OKS and a metric we call Shared Keypoint Rate (SKR). OKS is undefined for when the ground truth contains a keypoint that is not labeled in the detection keypoints. We thus make a small modification to the metric, where we only calculate the keypoint similarity between keypoints present in either of the annotations. Relying on this metric then also has a side-effect where the algorithm would attempt to minimize the number of shared joints as any difference between annotated joints would be excluded if one of them is not present. We thus combine it with SKR, which we define as the number of unique keypoints present in both annotations divided by the number of unique annotations in just the target annotation.

*`https://github.com/Vinno97/improved-mturk-keypoints-ui`

The pose solving algorithm gathers all annotations belonging to the same person and temporarily applies a selected transformation. It then cross-compares the *OKSSKR* (Equation (3.5)) for the transformed annotations against the non-transformed annotations and averages these scores. It then runs the same cross-comparison between the untransformed annotations and subtracts this from the previous scores. A positive value now indicates that applying the transformation improves its similarity towards its peer annotations, a negative value implies the opposite. We take the maximum of the scores and check if it is positive. If this is the case, we keep the transformation for this annotation and continue to the annotations for the next person.

$$\mathrm{OKS}(p, p') = \frac{\sum_i exp(-d(p_i, p'_i)^2/2s(p)^2k_i^2)\delta(p_{iv} > 0)}{\sum_i \delta(p_{iv} > 0)} \tag{3.2}$$

$$\mathrm{OKS}'(p, p') = \frac{\sum_i exp(-d(p_i, p'_i)^2/2s(p)^2k_i^2)\delta(p_{iv} > 0)\delta(p'_{iv} > 0)}{\sum_i \delta(p_{iv} > 0)\delta(p'_{iv} > 0)} \tag{3.3}$$

$$\mathrm{SKR}(p, p') = \frac{\sum_i \delta(p_{iv} > 0)\delta(p_{iv} > 0)}{\sum_i \delta(p'_{iv} > 0)} \tag{3.4}$$

$$\mathrm{OKSSKR}(p, p') = \mathrm{SKR}(p, p')\frac{\mathrm{OKS}(p, p') + \mathrm{OKS}(p', p)}{2} \tag{3.5}$$

$$\tag{3.6}$$

3.2 Google Images

The research questions of this thesis require a child-centered HPE dataset of sufficient size to (re)train modern deep learning models. The 322 poses from the previous section are not sufficient for this goal. We supplement this dataset with photos queried via Google Images. Images are collected via the process described in Figure 3.3.

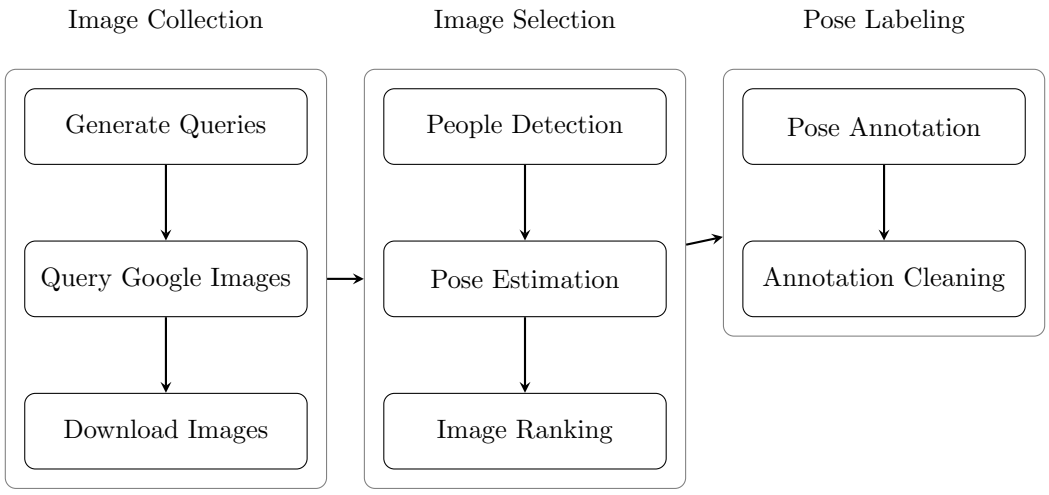


Figure 3.3: Flow diagram of the process for collecting the images from Google Images.

3.2.1 Obtaining images

Where the queries used in Section 3.1.1 were manually crafted, this section of the dataset uses automatically generated query expansions. By combining one of the 40 action labels from *Kinetikids* (Appendix A.2), with one of seven variations of "child" (Appendix C.1), using one of three stitching templates (Appendix C.2), we can generate a total of 861 different queries. Translating these to the 20 most commonly used languages on the web [90] (with Chinese translated into both traditional and simplified Chinese) and removing duplicate translation, finally yields 14248 different queries.

This process has two benefits when compared to the manually crafted queries. When querying either images on Google Images or videos on YouTube, the relatedness of results to the query decreases the further down the list one goes. This places a limit on the number of samples that can be retrieved per query. Running multiple variations of the same query greatly limits this effect [23]. Translating the queries also has another significant effect, namely that it provides data with a much greater geo-diversity.

Whilst querying Google, we use three "advanced search" constraints: we 1) only want images larger than 400 by 300 pixels, to filter out low-resolution images at the earliest possible stage. We 2) only want images that google classifies as a "photo" to reject cartoons. And we 3) only want images of the full-color type. This removes all images with transparent sections, which are unlikely to be normal photos of children doing the queried sports. After querying and deduplicating image URLs, we end up with 496193 images, which we convert to JPEG at the 95% quality setting.

3.2.2 Data Annotation

We use the same people detection and bounding-box filtering on these images as done in Section 3.1.2, though notably now not followed by the child filtering. Also different from the previous part of the dataset, we do not use AMT workers to annotate the images. Instead, we rely on labeling by students and staff of Utrecht University. We rely on just one annotation per person instead of averaging three annotations. As there is no need for grouping annotations here, we can also present annotators with uncropped versions of the source image. Annotators are free to chose which (not yet annotated) child they annotate. For future research, we also annotated whether or not the annotated child was interacting in some form with another person in the image. Some images contain many children, but little variance between poses. In these cases, only a select few of the subjects were annotated.

Figure 3.4: Example of one image throughout different rounds of annotation

3.3 Compiling Validation Sets

Using part of the newly gathered *Kinetikids-pose* to determine if training on child data (using *Kinetikids-pose*) improves performance would be problematic. Intricacies induced during data collection or annotation would logically be present in both the training as test split.

Instead, we compile two validation sets based on the 2017 COCO validation set. Both the images and the annotations are completely separate from *Kinetikids-pose*. We manually label 7256 of the 11004 annotated people in de dataset to be either "child", "adult", or "rejected". People without no labeled keypoints or whose bounding box is smaller than 50 pixels in height are automatically rejected. We also manually reject annotations that are not clearly visible or not clearly definable as a child or adult. We end up with 323 child annotations from 168 different images and 2857 adult annotations from 1287 different images.

As visualized in Table 3.3, both the child and (at this stage still) unfiltered adult subsets are easier to classify than the complete validation set. This can be explained by the fact many of the difficult-to-classify subjects are likewise difficult to estimate poses for. This is supported by the low AP scores for the rejected poses. This, however, also shows that we cannot directly compare the performance of off-the-shelf HPE algorithms between our subsets. The complexity of the subsets directly depends on what the human annotator thinks is "clearly" an adult or "clearly" a child.

	SimpleBaseline	HRNet	**Average**
COCO (val)	0.724	0.781	**0,753**
COCO Adult (unfiltered)	0.755	0.812	**0.784**
COCO Adult	0.803	0.857	**0.830**
COCO Child	0.763	0.826	**0.795**
COCO Rejected	0.529	0.601	**0.565**

Table 3.3: Baseline AP scores of off-the-shelf human pose estimators on each of the COCO-derived subsets.

3.3.1 Minimizing Selection Biases

Our goal is to compare the performance of off-the-shelf HPE algorithms between our labeled subsets. The manually filtered adult and child subsets are not, however, not directly comparable in their unprocessed form. To solve this, we algorithmically adjust the complexity of the annotations to minimize the effect of any selection biases.

We first note our adult set contains nearly a factor of nine more annotations than the child set. This gives room to filter out many difficult or easy annotations from this dataset that are too dissimilar to those found in the child dataset. Iterating over each annotation in the child split, we greedily select the most similar pose from the adult split based on solely pose metadata.

We use an algorithm that iteratively picks one child annotation and compares it against each of the adult annotations. It matches the child annotation with the most similar adult annotation and removes both annotations from our lists. When there are no more child annotations to choose from, the process is finished and we remain with a matched set of child and adult annotations. The child annotations are the same as those before the equalization algorithm. The adult annotations are now a filtered list of equal length to the list of child annotations and make up what we from now on refer to as "COCO Adult".

When comparing annotations, we do not want to directly compare poses or the contents of their corresponding images. Instead, we aim to equalize on pose metadata alone. We make the assumption that two poses with similarly sized bounding boxes and similar visible keypoints are also similarly difficult to label for a pose estimator. The algorithm does not consider information about the source images.

Formalized in Equation (3.13), the similarity metric is defined as the product of two binary Jaccard indexes (Equation (3.7)), one for each of the two types of visible keypoints, and two shape similarities that compare width (Equation (3.11) and height (Equation (3.11)). A and B are both matrices of shape $N \times 3$ where N is the number of keypoints and the three columns are the continuous x, y coordinates, and the discrete visibility v ($\{\mathcal{X}_{iv} \in \{0, 1, 2\}\}$). As the Jaccard index is only defined for for binary values, take one Jaccard index for where $\mathcal{X}_{iv} = 1$ and one for $\mathcal{X}_{iv} = 2$. J_ψ (Equation (3.8)) and $\Delta_\psi(\mathcal{Z})$ (Equation (3.9)) take an arbitrary matrix and together output the Jaccard index for where $\mathcal{X}_{iv} = \psi$. Lastly, $S_{width}(A, B)$ and $S_{height}(A, B)$ compare the bounding box sizes by dividing the smallest sized bounding box by the largest one along either the horizontal or vertical axis, respectively. $J(A, B)$ would result in a division by zero in case neither of the annotations contains labeled keypoints, though none of such annotations exist in these datasets.

$$J(A,B) = \frac{\sum_i A_{iv} B_{iv}}{\sum_i \min(A_{iv} + B_{iv}, 1)} \tag{3.7}$$

$$J_\psi(A,B) = J[\Delta_\psi(A), \Delta_\psi(B)] \tag{3.8}$$

$$\Delta_\psi(\mathcal{Z})_{iv} = \begin{Bmatrix} 1 & \text{if } \mathcal{Z}_{iv} = \psi \\ 0 & \text{if } \mathcal{Z}_{iv} \neq \psi \end{Bmatrix} \tag{3.9}$$

$$D_c(\mathcal{Z}) = \max_{i \in K} \mathcal{Z}_{ic} - \min_{i \in K} \mathcal{Z}_{ic} \tag{3.10}$$

$$S_{width}(A,B) = \frac{\min(D_x(A), D_x(B))}{\max(D_x(A), D_x(B))} \tag{3.11}$$

$$S_{height}(A,B) = \frac{\min(D_y(A), D_y(B))}{\max(D_y(A), D_y(B))} \tag{3.12}$$

$$S(A,B) = J_1(A,B) J_2(A,B) S_{width}(A,B) S_{height}(A,B) \tag{3.13}$$

3.4 Visualizing the data

All images in *Kinetikids-pose* were collected in the context of sporting activities, resulting in a wide variety poses. In this section, we present example annotations of *Kinetikids-pose* and visually compare the expressiveness of this dataset and the two COCO-derived validation sets. We finalize this section, and with that this chapter, by discussing the sizes of the gathered datasets.

In Figure 3.5, we present three randomly selected example images from the *Kinetikids-pose* training split. From left to right, the images were queried for "cartwheeling", "throwing frisbee" and "playing basketball", respectively. The children in the leftmost image notably do not perform the queried action. This data in this dataset is selected for HPE purposes irregardless of the action class. We thus place no restrictions that the images in the dataset have to contain any of the queried activities.

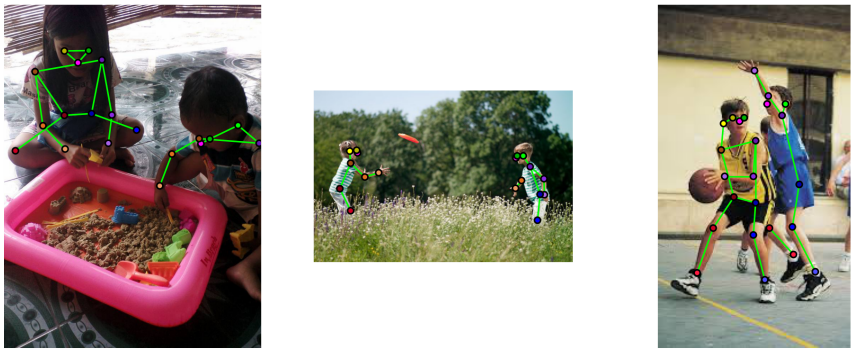
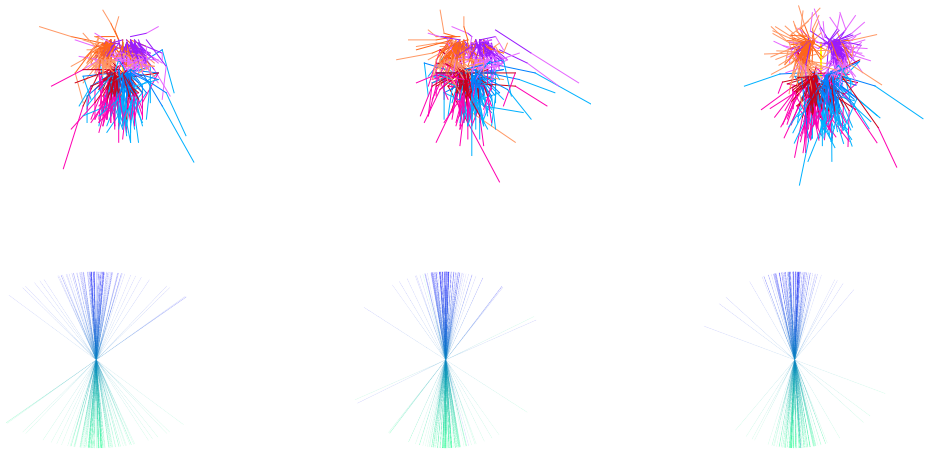


Figure 3.5: Example images with annotations from the *Kinetikids-pose* dataset. An extended version of this figure, also showing *COCO Adult* and *COCO Child*, is available in Appendix D.

Where Figure 3.5 visualizes the poses for three selected images, we also present aggregated visualization in Figure 3.6. These top images each superimpose 100 normalized poses on top of each other and serve to indicate the variety of poses. As the normalization rotates all poses upright, we also visualize their original rotations in the bottom row.

We finish this data analysis by presenting annotation statistics per compiled dataset in Table 3.4. We count the number of people as the number of people with one or more labeled keypoints. The number of images is counted likewise as images that contain at least one labeled person. Even though the validation set of COCO contains 5000 images and 11004 people, only 2346 images



(a) COCO Adult (b) COCO Child (c) Kinetikids

Figure 3.6: Visualization of 100 randomly selected poses from each dataset, filtered for poses with all shoulders and hips labeled. All poses are given a calculated "spine" as the line between the center of their shoulders and the center of their hips. The top row shows skeletal where all poses are rotated, translated, and scaled such their spines align into a single, vertical line. The bottom row shows the spines in their original orientations before the normalization step; blue indicates the top of the spine, teal indicates the bottom. More visualizations for different constraints can be found in Appendix D.

contain any of the 6352 with labeled keypoints. We also present statistics for the unfiltered version of COCO Adult, the rejected COCO annotations, the YouTube part of *Kinetikids-pose*, and the Google Images part.

Dataset (split)	Images	People	People per image	% labeled keypoints
COCO val (2017)	2346	6352	2.71	36.5
COCO Child	168	323	1.92	71.4
COCO Adult	274	323	1.18	73.6
Kinetikids (train)	851	1121	1.32	68.9
Kinetikids (val)	213	263	1.23	66.3
COCO Adult (unfiltered)	1278	2857	2.24	68.2
COCO Rejected	431	632	1.47	43.9
Kinetikids (YouTube)	219	319	1.46	59.4
Kinetikids (G-Images)	845	1065	1.26	71.1

Table 3.4: Annotation statistics per dataset.

Chapter 4

Experiment and Results

In this chapter, we describe the setup and the subsequent results of our experiments. The chapter starts out by defining the experimental setup and how the results will be compared. The next section describes the reasoning for the selection of our baseline model, followed by an description of how this model was finetuned. Finally, we will present the results of the defined experiments.

4.1 Experimental setup

We will compare the performance of two different pose estimators between the newly compiled *COCO Child* and *COCO Adult* datasets. Via this comparison, we aim to to deduct if the poses from either of the two datasets show to be more challenging to estimate than those from the other set. The samples are taken from the same superset and care was taken to minimize differences in annotation sizes and visible keypoints, this should thus allow for fair comparison of performance indicators.

As our models, we select *SimpleBaseline*, in its ResNet-50 variant with an image size of 256×192, and *HRNet*, in its W48 variant with an image size of 384×288. For the remainder of this chapter, both *SimpleBaseline* and *HRNet* will refer to these specific configurations. These are the most commonly used reference implementations and incidentally the highest performing reference HRNet variant and the lowest-performing reference *SimpleBaseline* variant.

We also finetune *SimpleBaseline* on *Kinetikids-pose* and test the resulting model on both *COCO Child* and *COCO Adult*. Afterward, we will compare the results against the previously measured baseline scores, produced by the non-finetuned model. The hypothesis is that since children are underrepresented in the tested COCO dataset, finetuning it on *Kinetikids-pose* will yield higher estimation accuracy on *COCO Child*. The baseline model has been trained until convergence on the COCO dataset. It is thus also likely that its performance will not rise much further during the finetuning phase. We thus also anticipate an outcome where training on *Kinetikids-pose* results in a performance regression on *COCO Child*. In such case, we analyze if there is also a performance regression on *COCO Adult* and, if so, if this regression is then statistically greater than that on *COCO Child*.

4.1.1 Evaluation Metrics

This chapter contains several performance evaluations, across differing levels of abstraction. Individual keypoint predictions are compared using the Keypoint Score (KS), defined as the euclidean distance scaled by the area of the ground truth bounding box and multiplied by a per-keypoint scaling constant. color=blue]I moved all texts that refer to the KS and OKS to the discussions section. I should probably move part of this section to there in a next draft. This scaling constant is based on average per-keypoint human annotation errors and penalizes errors of precise keypoints like eyes stronger than, for example, hip keypoints. Individual poses are evaluated using the OKS, defined as the mean of the KS scores across only the labeled keypoints. Lastly, the main metric used to compare performance across datasets is the Average Precision (AP). This metric is defined equal to the AP used as the main challenge metric of COCO. Precision scores are calculated as the

proportion of poses with an OKS greater than a certain threshold. The AP takes the average the precision scores across the thresholds 0.05, 0.5 and 0.95. $AP_{0.5}$ and $AP_{0.75}$ refer to the precision for the thresholds 0.5 and 0.75, respectively.

Statistical analysis on same-dataset performance measures for different models is performed using a paired t-test. Performance measures for different datasets are subsequently compared using an unpaired t-test.

4.1.2 Baseline Model

The *Kinetikids-pose* dataset contains many images that contain more people than annotated poses. It contains only keypoint annotation for child subjects, irregardless of if adult subjects are present in the image. These unlabeled subjects are unmasked and their joints would be considered false positives for any algorithm that analyzes the image globally. This therefore limits the possible models to the ones that can work with sparsely annotated images. Bottom-up models are unlikely to converge optimally with this dataset due to this limitation and can thus all be eliminated. Top-down models only look at a specific region around the provided bounding box for a person and are thus not affected by the unannotated people.

At the moment of writing, to our knowledge, eight of the twelve top performing top-down models on the COCO keypoints dataset are variations of either SimpleBaseline or HRNet (see Table 4.1).color=blue]Got this data from paperswithcode.com. The COCO challenge leaderboard contains many more slightly different variations on HRNet. Some of the highest performing models from the paperswithcode list don't actually participate in the challenge, however. Both models have proven to be reliable bases for further experimentation and are thus good baseline models. We choose to finetune SimpleBaseline, as its straightforward architecture makes it easier to experiment on. All tests are performed using a ResNet-50 model with an input image size of 256×192 which is pretrained on COCO.

Model	Variety	Base Model	Test AP	Source
UDP-Pose-PSA	384×288	HRNet	79.5	Liu et al. [56]
UDP-Pose-PSA	256×192	HRNet	78.9	Liu et al. [56]
EvoPose2D-L	-	EvoPose	78.9	McNally et al. [61]
PoseFix	-	SimpleBaseline	76.7	Moon, Chang, and Lee [62]
DarkPose	-	HRNet	76.2	Zhang et al. [96]
MSPN	-	CPN	76.2	Luo et al. [58]
HRNet	W48	HRNet	76.2	Sun et al. [86]
CPN+	-	CPN	73.0	Chen et al. [17]
PNFS	-	SimpleBaseline	70.9	Yang, Yang, and Cui [95]
Mask R-CNN	-	Mask R-CNN	66.5	He et al. [34]
HRNet	W32	HRNet	75.8 *	Sun et al. [86]
SimpleBaseline	ResNet-50	SimpleBaseline	72.2 *	Xiao, Wu, and Wei [94]

Table 4.1: Top-performing top-down HPE models (as of writing). Models with a * behind the AP scores are only tested on the COCO validation set, instead of the COCO test set.

4.1.3 Finetuning SimpleBaseline

We start with a SimpleBaseline model, pretrained on COCO Keypoints 2017[*], which we aim to finetune on *Kinetikids-pose*.

[*]Model downloaded from `https://github.com/microsoft/human-pose-estimation.pytorch`, weights downloaded from `https://drive.google.com/file/d/1DIhf0DoyHjTkk_14BshTAdbgaa9ApnET`

A common strategy for transfer learning is to only alter the weights of certain layers at the end of the model, whilst "freezing" the weights of the rest. The first layers then act as static "feature extractors", whilst the layers that further transform these features are allowed to learn. This effectively results in a simpler trainable sub-model with less potential for overfitting and quicker convergence. Adversely, every additional frozen layer reduces the model's ability to learn new patterns.

SimpleBaseline consists of three major sections: a ResNet-50 CNN feature extractor, a sequence of three deconvolutional layers to upsample the ResNet features , and a 1×1 convolutional layer that converts the deconvolutional features into the final heatmaps. The feature extractor is a standard Resnet-50 model without the classification layer. This means it uses an initial 7×7 convolution with a stride of 2 to reduce the initial dimensionality, followed by four ResNet bottleneck layers that convert the image to a $6 \times 8 \times 2048$ feature map. Each ResNet layer halves the resolution.

We identify the first convolutional layer (with corresponding batch normalization layer), each of the ResNet bottleneck layers, the deconvolutional section and the final 1×1 layer as the main stages of this model. Each of these stages observes a different level of abstraction and resolution, except the deconvolutional section that enlarges the observed features three times in one stage. We test the effect of freezing each of these main stages of this model on its potential for learning and overfitting in Figure 4.1. From this, we choose to freeze all downsampling convolutional layers, only finetuning the decoding section of SimpleBaseline.

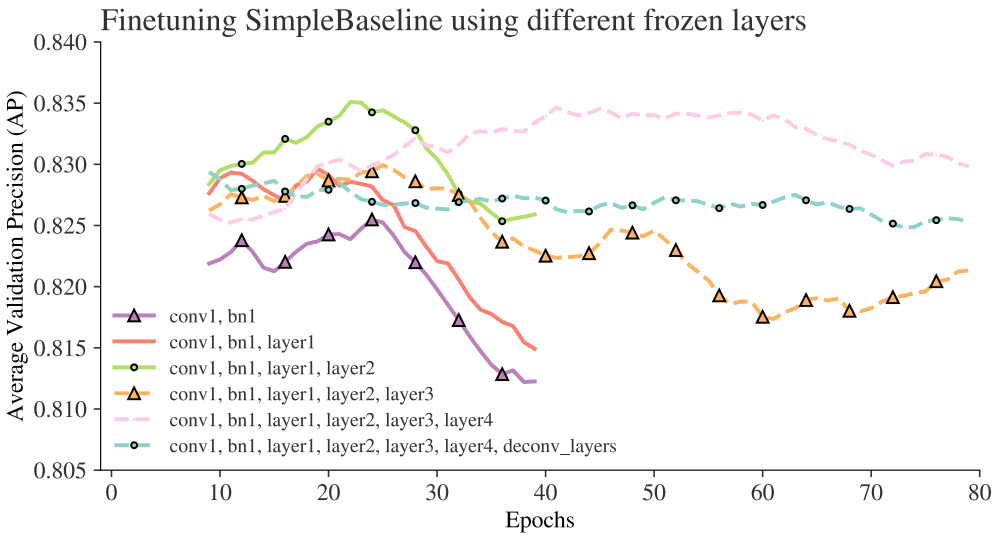


Figure 4.1: Learning curves for finetuning SimpleBaseline on *Kinetikids-pose* using different frozen layers. Training for the first three models was stopped halfway at 40 epochs. All curves are smoothed with a rolling average of window size 10. Layer naming follows the internal naming in the SimpleBaseline code; layer{1...4} resemble one ResNet module each.

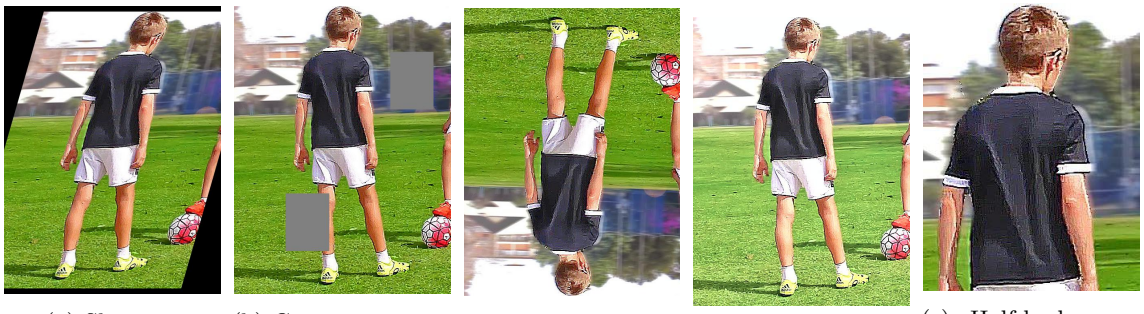
To further reduce overfitting potential, we test various methods for data augmentation. As the convolutional layers are frozen, we focus on augmentations whose effects should propagate throughout these layers and subsequently affect the trainable upsampling layers.

We visualize the effects of the selected augmentations on the finetuning performance in Figure 4.3. We test vertical flipping, half-body augmentation, shearing, rectangular cutout [22], and random brightness jitter. All tests also use the base augmentations of random rotation ($\pm40\%$), random scaling ($\pm30\%$), and horizontal flipping (50% chance) used by the authors of SimpleBaseline in addition to the tested augmentation.

Shearing displaces all pixels horizontally by a set amount compared to the previous row of pixels (Figure 4.2a). Rectangular cutout masks random rectangles of the source image with a constant value (Figure 4.2b). With vertical flipping, samples are flipped upside down 50% of the time (Figure 4.3c). Random brightness jitter multiplies the brighness of the image by a randomly

chosen value within a range (Figure 4.2d). And finally half-body augmentation crops the sample to either the top or bottom half of a person's body when there are sufficient keypoints present (Figure 4.3e).

With the exception of vertical flipping, none of the tested augmentations greatly affect SimpleBaseline's ability to generalize on *Kinetikids-pose*. Shearing and cutout seem to have minimal effect, though half-body augmentation and 12.5% brightness jitter do seem to improve generalization performance. We decide on brightness jitter (±12.5%) and half-body augmentation (30% chance) as the augmentations for the final model.



(a) Shearing
16°

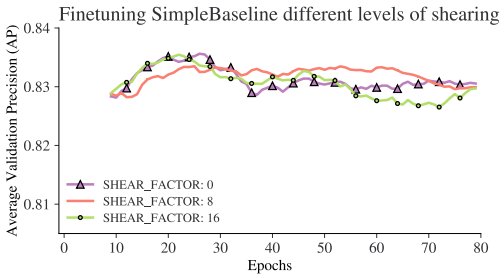
(b) Cutout
2 Iterations

(c) Vertical Flip

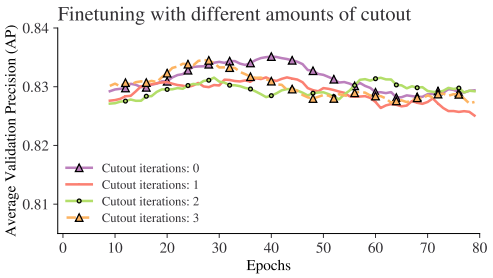
(d) Brightness Jitter
×1.25

(e) Half-body augmentation
Upper body

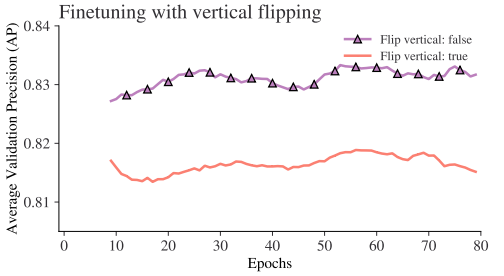
Figure 4.2: Example images of augmentations



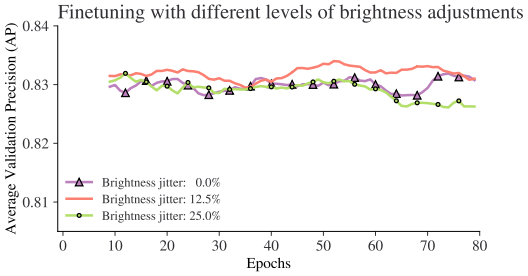
(a) Shearing



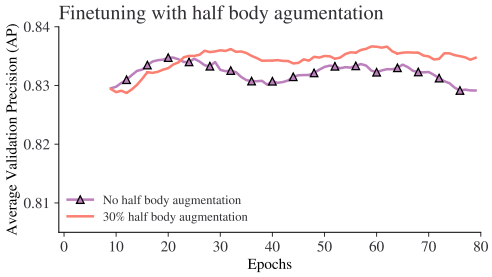
(b) Cutout



(c) Vertical Flip



(d) Brightness Jitter



(e) Half-body augmentation

Figure 4.3: Learning curves for finetuning SimpleBaseline on *Kinetikids-pose* with different data augmentation strategies. All curves are with a rolling average with window size 10.

4.2 Model evaluation

All performance discussed in tests discussed in Section 4.1 are summarized in Table 4.2.

SimpleBaseline and HRNet both perform better on *COCO Adult* than they do on *COCO Child*. SimpleBaseline achieves an AP of 0.803 and 0.763 on them, respectively. HRNet 0.857 and 0.826. Comparing the combined scores per dataset results using an independent t-test results in a P of 0.0018, indicating the difference is significant. This is smaller than 0.05, meaning these differences are statistically significant.

	COCO Child			COCO Adult			Kinetikids		
Model	AP	$AP_{0.5}$	$AP_{0.75}$	AP	$AP_{0.5}$	$AP_{0.75}$	AP	$AP_{0.5}$	$AP_{0.75}$
HRNet	0.826	0.980	0.888	0.847	0.964	0.919	-	-	-
Baseline SB	0.763	0.959	0.845	0.803	0.966	0.898	0.836	0.973	0.872
Finetuned SB	0.722	0.949	0.815	0.771	0.956	0.882	0.838	0.972	0.880

Table 4.2: Model performances across Kinetikids, COCO Child, and COCO Adult.

After finetuning on *Kinetikids-pose*, the SimpleBaseline model performs marginally better on the corresponding validation split. Visualized in Figure 4.4 and also shown in Table 4.2, the finetuned model edges out the baseline model with an AP improvement of 0.002. This comes at the cost of an AP drop of 0.032 for *COCO Adult* and 0.042 for *COCO Child*. An unpaired t-test between the OKS differences provides a P of 0.22, also not significant.

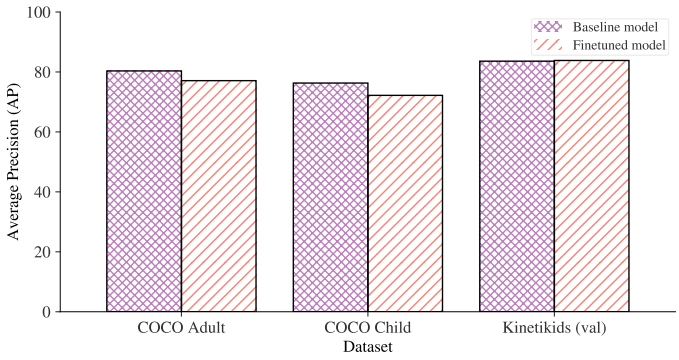


Figure 4.4: AP scores

Chapter 5

Discussion

This chapter we will analyze the results of Chapter 4 and discuss them in relation to the research questions from Section 1.4. We also analyze the performance of the finetuned model and explore several explanations as to why the chosen model was unable to significantly improve its performance on *Kinetikids-pose*. We explore limitations in both the model and the dataset.

5.1 Experiment Evaluation

In Section 4.2, we see that both HRNet and SimpleBaseline show a lower accuracy on *COCO Child* than on *COCO Adult*. With a P of 0.0018, the difference in results is also statistically significant. From this we can thus safely conclude there is indeed be a decrease in performance on children when compared to adults, hereby answering RQ 1..

We also see that the finetuned performance of SimpleBaseline on *Kinetikids-pose* only marginally edges out the baseline performance on this dataset. Finetuning, however, considerably reduces the performance on both COCO sets. There is thus no need for validating if any increase in performance is significant, as was proposed in Section 4.1. The performance regression of *COCO Child* is slightly greater than that of *COCO Adult*. This difference is, however, not significant with a P of 0.22. RQ 2. questions if training on child data improves the performance of a model on child poses. From these results, we are not able to establish such a relation.

The main research question for this thesis, "Is the performance of current SOTA pose estimation on children limited by the adult-biases of the datasets that they are trained on?", depends on both of the sub research questions to be true for it to be true. The failure to confirm RQ 2. thus also leaves us unable to prove this question.

5.1.1 Model Limitations

A paired t-test between the validation OKS scores for the baseline model and the finetuned model gives a P of 0.06 This indicates that finetuning the model did not succeed in improving upon the evaluation metric in a statistically significant amount. In this section, we thus explore mechanics that could have contributed to this lacking performance gain.

Frozen Layers

In finetuning SimpleBaseline, we decided to freeze all initial convolutional layers. This was done in an effort to prevent overfitting. These initial layers, however, are responsible for most of the pose perception. Deeper levels of the model receive increasingly more localized information, shown in Figure 5.1. This also easily visualizes why finetuning only the final layer had little to no effect. The four ResNet modules compress the image down to a 6×8 image with 2048 values per pixel. All image processing has already happened at this stage and the compressed spatial features are optimized to contain only the required pose information to create a higher resolution heatmap.

By freezing all the convolutional layers, we impede the model from learning any new lower-level visual patterns. Unfreezing any more layers, however, proved to result in quick overfitting. Perhaps a baseline model designed when large pose datasets were less prevalent (like Tompson et al. [88]) may have been more perceptible to finetuning with the limited available data.

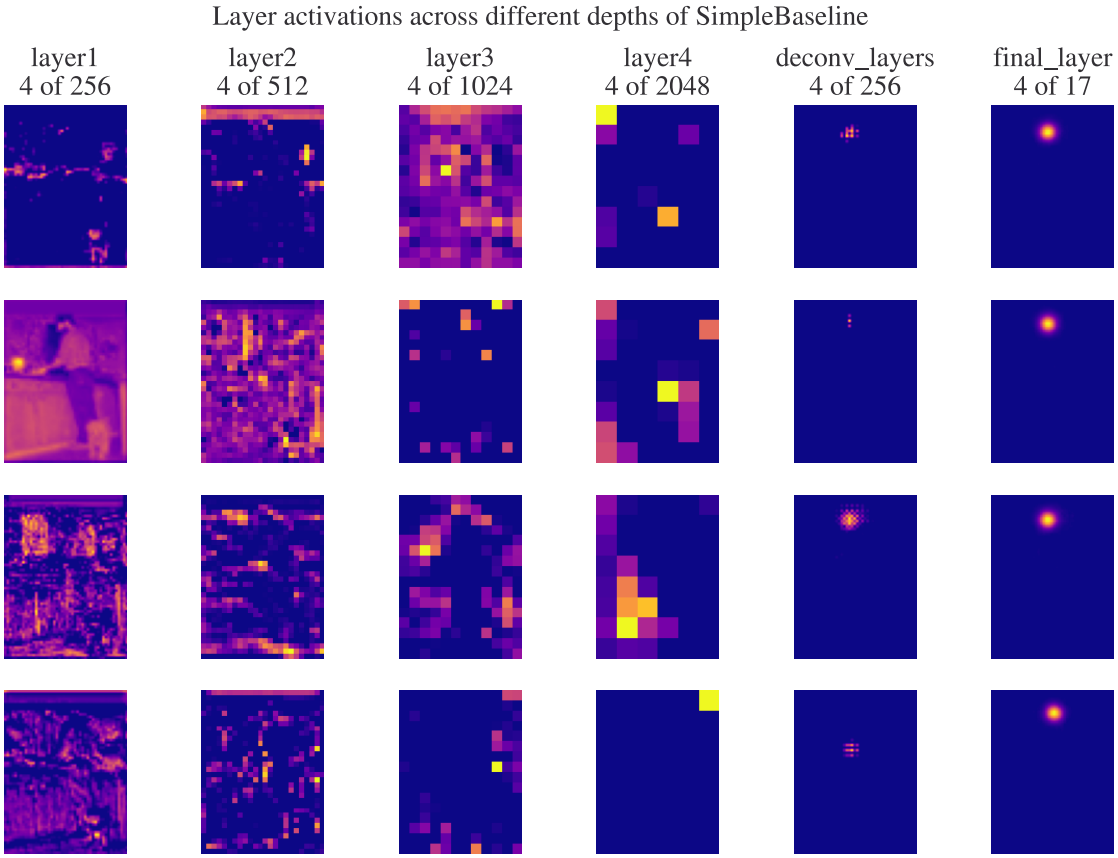


Figure 5.1: Example layer activation across different depths of SimpleBaseline. For each stage, we visualize the activations of four channels. Note that channels visualized on the same row do not have to focus on the same sections of the image as there is no ordering to which channel focus on which features.

Converged Model

Another reason that finetuning the baseline model on *Kinetikids-pose* had minimum effect on its performance on the *Kinetikids-pose* validation split, might be because the model was already pretrained to convergence on the full COCO training set of 200,000 + images. The baseline model is already very close to its maximum performance level. Transfer learning attempts with similar data to which it was trained on to convergence can thus only have a limited effect.

A possible alternative methodology could be to train the SimpleBaseline model with the complete COCO train set for a certain fraction of the epochs required for convergence. This new baseline model would then have a passable but not exceptional HPE accuracy. It should, however, be more susceptible to accuracy improvements via transfer learning.

The 1121 poses in our training set were unable to have a significant impact on the fully converged model's performance, but might have been able to have more of an effect on this hypothetical unconverged version. We also did not test if training on an adult training set would have the same negative effect on the model's performance. Or if this set would disproportionately affect children.

Training to Extremities

Visualized in Figure 5.2, we see that finetuning on *Kinetikids-pose* has resulted in the model being able to label more people with a very high OKS (OKS $\rightarrow$ 1) score. Meanwhile, there are also more poses it scores a very low OKS (OKS $\rightarrow$ 0) score on. This can indicate the model becoming better in estimating the poses it is already good at, whilst becoming worse at the more difficult poses.

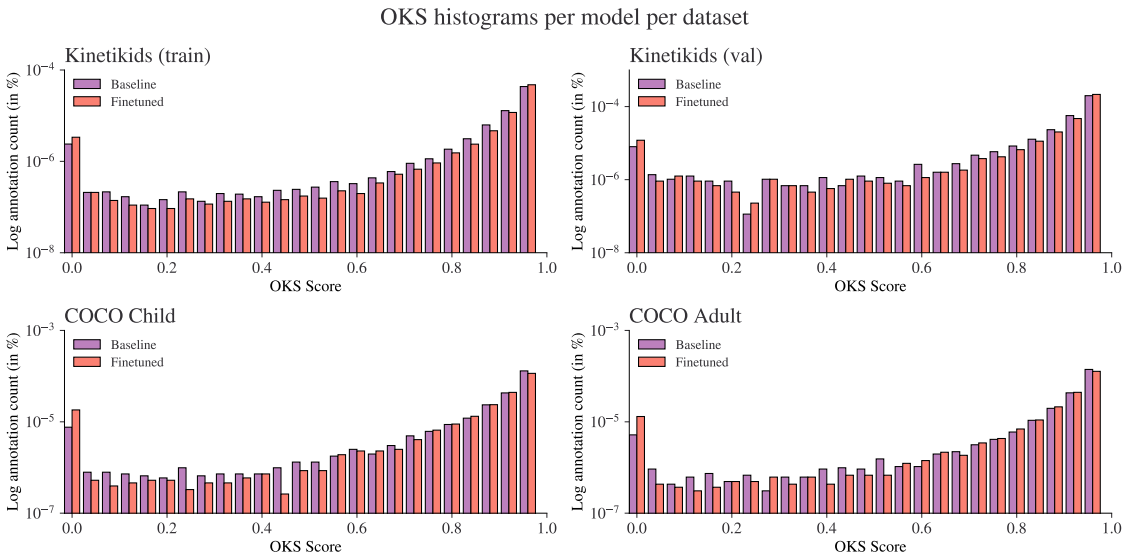
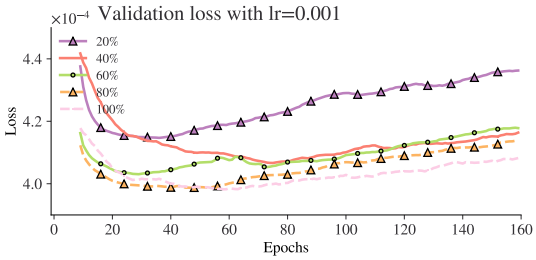


Figure 5.2: Histograms of OKS values for each dataset before and after finetuning.

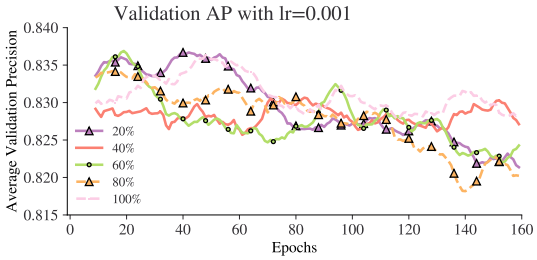
5.1.2 Dataset Size

Finetuning a pretrained deep learning model requires fewer samples than when one would train that same model from scratch. It is, however, still desired to have a dataset of multiple thousands of samples for such a task. The COCO train split contains 139,486 poses and still techniques like OpenPose include additional sources like 40,522 poses of MPII to augment the boost of training samples size. The training split of *Kinetikids-pose* consists of just 1121 poses from 851 source images.

To confirm if the performance of our finetuned model was limited by the size of the training set, we attempt to finetune the same baseline model using 20%, 40%, 60%, 80% and 100% of the *Kinetikids-pose* data. A consistent increase in localization performance per increased training size, would indicate that our model would benefit from more data. Figure 5.3, however, does not show such a relation. This would indicate the model is either limited elsewhere, or the amount of data is so little that the decrease in data does not demonstrate a significant decrease in performance.



(a) Loss curve



(b) Predictions by the finetuned model

Figure 5.3: Training AP and loss for finetuning the baseline model with 20%, 40%, 60%, 80% and 100% of the *Kinetikids-pose* data. Both curves are smoothed with a rolling average with window size 10.

5.2 Dataset

Finetuning on *Kinetikids-pose* only marginally improved our model's accuracy on *Kinetikids-pose*, but noticeably regressed the performance on both *COCO Child* and *COCO Adult*. This indicates a difference between the newly collected data and the COCO reference data in any of or a combination of a) the visual properties of the images, b) the manner of posing, or c) the manner of annotation. In this section, we investigate these potential causes.

5.2.1 t-SNE

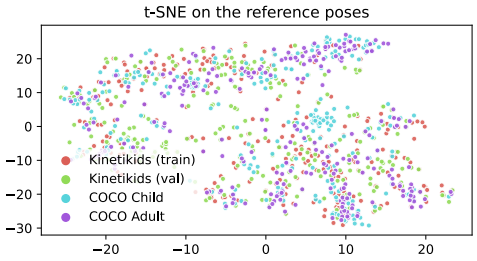
As an initial assessment, we use t-SNE [60] to plot inter-pose differences between datasets. This dimensionality reduction method uses statistical optimization to map higher dimensional data to lower dimension whilst maintaining relative distances between samples. We select equally sized random subsets for each dataset and project normalized x, y, v values for the joints of all poses onto a 2D space. We do this for the ground truth poses (Figure 5.4a), baseline predictions (Figure 5.4b) and predictions by the finetuned model (Figure 5.4c). None of the plots show clear seperated clusters between datasets, indicating the pose distributions to be very similar to each other.

5.2.2 Labeling Differences

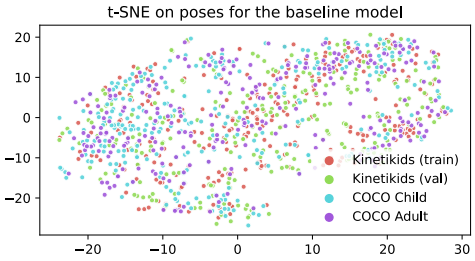
Plotting the per-keypoint difference in KS between the baseline and finetuned models (Figure 5.5) provides further insight into where the finetuned model makes different predictions compared to the baseline model. Finetuning on *Kinetikids-pose* slightly improves the model's accuracy on the hip and leg keypoints for this dataset. At the same time, the model's accuracy on the COCO hip and leg keypoints significantly decreases. This can be the result of (slightly) different manners of annotating these keypoints. Facial landmarks like the eyes, the keypoints with the smallest performance difference, are unambiguously recognizable, whilst the hips, the keypoints with the largest performance difference, can be more difficult to place precisely.

Figure 5.6 shows the number of poses per dataset that contain a certain percentage of labeled poses, a percentage of 100% means all 17 possible keypoints are labeled. From these plots, we can see that the COCO validation sets contain comparatively more poses with a higher number of labeled keypoints than those from *Kinetikids-pose*. The *Kinetikids-pose* poses, in turn, show more poses with 50%-80% labeled keypoints. This figure also shows that *COCO Adult* contains comparatively less fully-labeled poses than *COCO Child*. This is in spite of the amount of visible keypoints being part of the similarity metric used to remove selection biases in Section 3.3.1. From Figure 5.7 we can infer that *COCO Child* also contains comparatively more poses where the facial keypoints are annotated, especially compared to *Kinetikids-pose*.

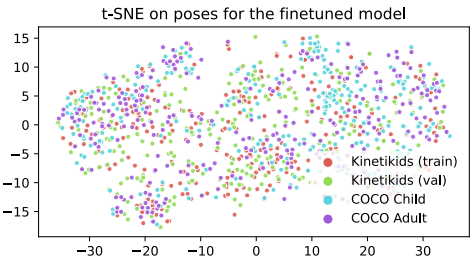
It is difficult to directly compare poses between datasets. These plots, however, do provide some insight. The training split of *Kinetikids-pose*contains comparatively less labeled keypoints for the face than for the rest of the body. It also has a noticeably smaller performance regression



(a) T-SNE on reference poses.



(b) t-SNE on poses predicted by the reference model.



(c) T-SNE on poses predicted by the finetuned model.

Figure 5.4: T-SNE visualizations of equally sizes random subsets of the different datasets. All pose annotations are normalized such that the center of their Point-to-Point (PTP) bounding boxes are centered around $(0, 0)$ and have a height of 1.

on the facial keypoints of the COCO datasets in Figure 5.5. SimpleBaseline ignores unlabeled keypoints during the loss calculation, meaning low performance regression could be just the result of less update steps. This conclusion, however, does not explain the reason for any of the other keypoints with less of a performance regression is thus at best a partial explanation. For further conclusions, more analysis is required.

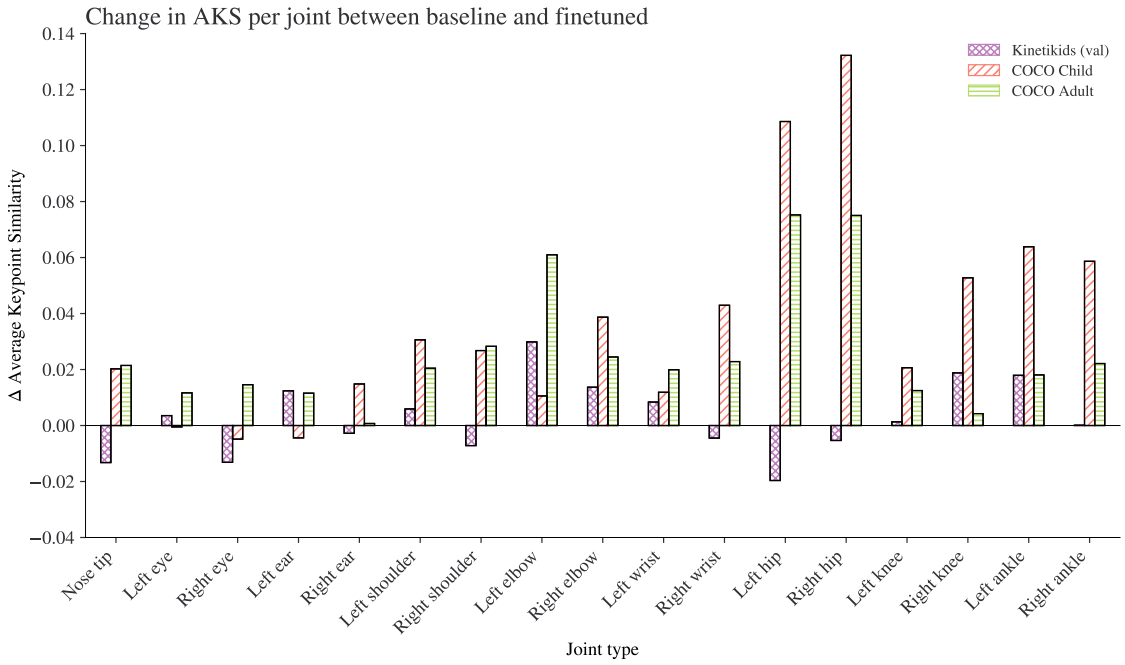


Figure 5.5: Per-joint KS changes averaged for Kinetikids, COCO Child, and COCO Adult.

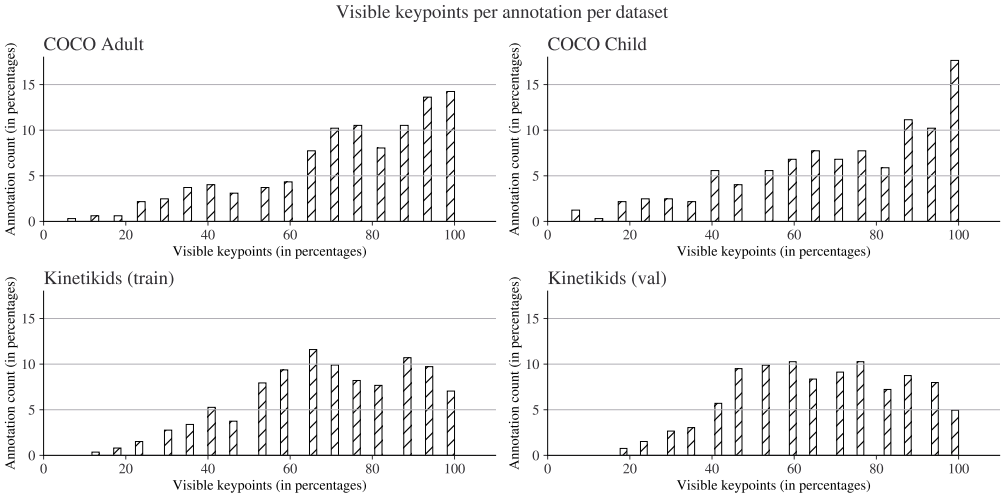


Figure 5.6: Histograms of visible keypoint percentage per person per dataset.

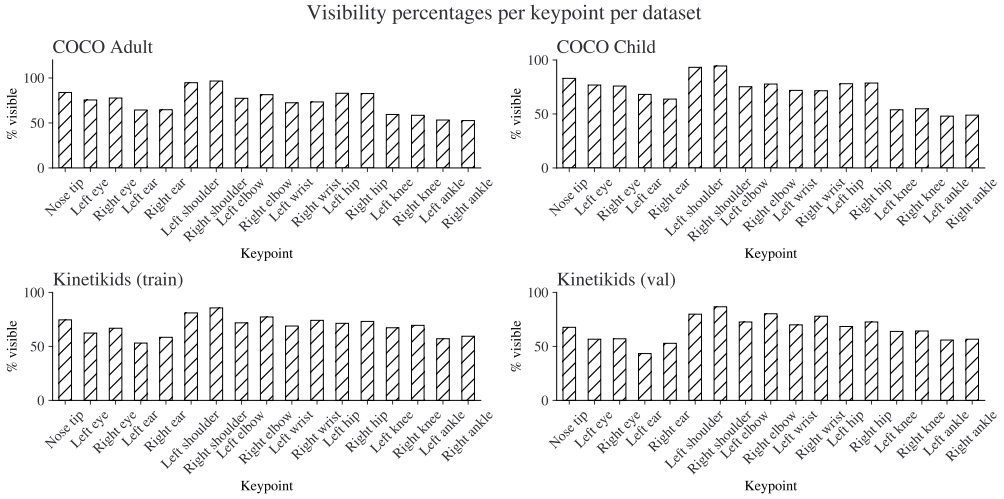


Figure 5.7: Keypoint occurrences per dataset.

Chapter 6

Conclusions

This thesis aimed to explore the presence and effects of adult biases in HPE datasets for child pose estimation. We created a new HPE dataset of children and filtered the validation set of COCO to create child-specific and adult-specific subsets.

In Chapter 2, we found that many of the current HPE datasets have an either advertent or inadvertent underrepresentation of children that can be directly linked to the content type they were compiled from, e.g. adult Hollywood actors. Also datasets without this immediate content bias contain a representational bias against children. Whilst filtering for *COCO Child* and *COCO Adult* in Chapter 3, we found that children make up merely 8% of the annotated people in COCO compared to the 75% of adults. The remaining 17% being people too unclear to classify.

This work provides strong evidence that child poses are more difficult to estimate for modern HPE models than those of adults. This concurs with the assumption that this observed bias also translates into a performance degradation for child pose estimation. This thesis, however, did not succeed in demonstrating if training a model on additional child data indeed improves the HPE accuracy on children for said model. It also does not prove the absence of such a relation. We thereby believe there is still valid reason to presume child pose estimation performance to be limited by predominantly adult-focussed datasets.

Limitations

In our finetuning experiments, we limited the learning ability of our model to prevent overfitting. We found this to be necessary when finetuning the model with our amount of training data. It may also inadvertently have limited the trainability of the model to the point where it was unable to sufficiently adapt its internal concept of the human body to that of children. Our follow-up experiments demonstrated that decreasing the amount of training data did not further lower the finetuning performance. This suggests that our current model was not data-limited in its training capacity. With a larger training set, however, there would be no (or less of a) need to freeze sections of our model. This would have allowed us to finetune SimpleBaseline in its entirety – without limiting its training potential. Selecting different settings for data augmentation and hyperparameters had equally little effect on the performance of the model. This could could also be contributable to the learning impairment we emposed on the model.

We also only attempted to finetune a single HPE model, instead of multiple models of different architectures. This effectively limits our reseach to answering if pose estimation on children via Simplebaseline is limited by adult-biased datasets, instead of answering if "SOTA pose estimation on children" is limited by these biases. The chosen model was also already pretrained to convergence on COCO. The same dataset from which we sample our validation datasets, and one that is close in domain to *Kinetikids-pose*. This limits any possible improvements whilst finetuning.

Lastly, the manner in which the finetuned model regressed in performance on the COCO datasets, also indicates a possible disparity between how keypoints are labeled in *Kinetikids-pose* when compared to COCO. We intended to create a COCO-like dataset containing solely children, though such a disparity further limits our ability to draw conclusions from the finetuning results.

Future work

This thesis uses a new dataset to finetune a HPE model on children; Sciortino et al. [83] used a new dataset to determine a difference in HPE accuracy between adults and children. Both works suffer from the fact that, unless utmost care is taken to prevent this, all datasets contain differences in pose complexity and/or labeling characteristics. Future work would benefit from instead using an existing large-scale HPE dataset, such as COCO, as a base and filter it in a manner as to how *COCO Child* and *COCO Adult* were constructed. This could be a comparatively simple process that would result in two large subsets without differences in data biases or method of labeling to the original dataset. The larger scale of these datasets would also enable training of HPE models without having to undermine their trainability by freezing too many layers.

Finally, the measured difference in pose estimation performance between adults and children, whilst measurable, is not substantial compared to inter-model performance differences. Considering the current techniques and data, we believe that if a specific child pose estimation task requires greater precision, model improvements should be prioritized over data improvements.

Acronyms

AMT Amazon Mechanical Turk. 17, 23, 24, 26

AP Average Precision. 30, 31, 35

BBC British Broadcasting Corporation. 17

BSL British Sign Language. 17, 18

CNN Convolutional Neural Network. 7, 10–16, 32, 46

CPN Cascaded Pyramid Network. 12

DSNT Differentiable Spatial to Numerical Transform. 9

HPE Human Pose Estimation. 1, 3–8, 10, 12–19, 23, 25–28, 31, 37, 43, 44

ILP Integer Linear Program(ming). 14, 15, *Glossary:* integer linear programming

KS Keypoint Score. 30, 39, 41

NN Neural Network. 6

OKS Object Keypoint Score. 23, 24, 30, 31, 38

PAF Part-Affinity Field. 15, 16

POI Point of Interset. 6, 46

PTP Point-to-Point. 40

RGB-D RGB-Depth. 3

RoI Region of Interest. 14

RPM Region-Proposal Model. 13, 14

SKR Shared Keypoint Rate. 24

SOTA State Of The Art. 1, 3–6, 8, 13, 14, 17, 23, 36, 43

Glossary

compendium A comprehensive collection of something. 17

Convolutional Neural Network A class of neural networks for spatial data that work by moving trainable filters over the spacial data. 7, 45

encoder-decoder model A neural network architecture consisting of two major components: an *encoder* component that condenses the input into a dense feature tensor and a *decoder* component that creates a new representation from the condensed repersentation.. 11

integer linear programming A linear mathematical model where all variables are contrained to be integers. 14

integral regression The process of regressing coordinates via a the soft *argmax* approximation on a heatmap. 9

keypoint A Point of Interset (POI) on an image. In the context of this thesis, it most oftenly refers to the location of human joints in an image. 6, 9

R-CNN From "Regions with CNN features", an object detection algorithm that creates (2000) proposal regions an classifies them with a CNN. 13, 14

receptive field Field of pixels that affect a pixel in a later feature map pixel in a . 10

regression Predicting a continuous value from a set of features. 46

Bibliography

[1] Naimish Agarwal, Artus Krohn-Grimberghe, and Ranjana Vyas. *Facial Key Points Detection Using Deep Convolutional Neural Network - NaimishNet*. Oct. 3, 2017. arXiv: 1710.00977 [cs, stat]. URL: http://arxiv.org/abs/1710.00977 (visited on 01/28/2021).

[2] Barbara E. Ainsworth et al. "Compendium of Physical Activities: An Update of Activity Codes and MET Intensities:" in: *Medicine & Science in Sports & Exercise* 32 (Supplement Sept. 2000), S498–S516. ISSN: 0195-9131. DOI: 10.1097/00005768-200009001-00009.

[3] Mykhaylo Andriluka et al. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, June 2014, pp. 3686–3693. ISBN: 978-1-4799-5118-5. DOI: 10.1109/cvpr.2014.471.

[4] Mykhaylo Andriluka et al. "PoseTrack: A Benchmark for Human Pose Estimation and Tracking." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 5167–5176. ISBN: 978-1-5386-6420-9. DOI: 10.1109/cvpr.2018.00542.

[5] Can Basaran et al. "Classifying Children with 3D Depth Cameras for Enabling Children's Safety Applications." In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*. Seattle, Washington: ACM Press, 2014, pp. 343–347. ISBN: 978-1-4503-2968-2. DOI: 10.1145/2632048.2636074.

[6] Lubomir Bourdev and Jitendra Malik. "Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations." In: *2009 IEEE 12th International Conference on Computer Vision*. Kyoto: IEEE, Sept. 2009, pp. 1365–1372. ISBN: 978-1-4244-4420-5. DOI: 10.1109/iccv.2009.5459303.

[7] Tom B. Brown et al. *Language Models Are Few-Shot Learners*. July 22, 2020. arXiv: 2005.14165 [cs]. URL: http://arxiv.org/abs/2005.14165 (visited on 02/01/2021).

[8] Adrian Bulat et al. *Toward Fast and Accurate Human Pose Estimation via Soft-Gated Skip Connections*. Feb. 25, 2020. arXiv: 2002.11098 [cs]. URL: http://arxiv.org/abs/2002.11098 (visited on 01/20/2021).

[9] Z. Cao et al. "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 1302–1310. DOI: 10.1109/CVPR.2017.143.

[10] Z. Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (Jan. 2021), pp. 172–186. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2019.2929257.

[11] J. Carreira et al. "Human Pose Estimation with Iterative Error Feedback." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 4733–4742. DOI: 10.1109/CVPR.2016.512.

[12] Brandon Castellano. "PySceneDetect." In: *GitHub* (2017). URL: https://github.com/Breakthrough/PySceneDetect.

[13] James Charles et al. "Domain Adaptation for Upper Body Pose Tracking in Signed TV Broadcasts." In: *Procedings of the British Machine Vision Conference 2013*. Bristol: British Machine Vision Association, 2013, pp. 47.1–47.11. ISBN: 978-1-901725-49-0. DOI: 10.5244/C.27.47.

[14] James Charles et al. "Automatic and Efficient Human Pose Estimation for Sign Language Videos." In: *International Journal of Computer Vision* 110.1 (Oct. 2014), pp. 70–90. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-013-0672-6.

[15] James Charles et al. "Personalizing Human Video Pose Estimation." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 3063–3072. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.334.

[16] Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu. "Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval." In: *Computer Vision – ECCV 2014*. Vol. 8694. Cham: Springer International Publishing, 2014, pp. 768–783. ISBN: 978-3-319-10598-7 978-3-319-10599-4. DOI: 10.1007/978-3-319-10599-4_49.

[17] Y. Chen et al. "Cascaded Pyramid Network for Multi-Person Pose Estimation." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 7103–7112. DOI: 10.1109/CVPR.2018.00742.

[18] Yucheng Chen, Yingli Tian, and Mingyi He. "Monocular Human Pose Estimation: A Survey of Deep Learning-Based Methods." In: *Computer Vision and Image Understanding* 192 (Mar. 2020), p. 102897. ISSN: 10773142. DOI: 10.1016/j.cviu.2019.102897. arXiv: 2006.01423.

[19] *COCO Leader Board*. URL: https://cocodataset.org/#keypoints-leaderboard.

[20] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database." In: *CVPR09*. 2009. DOI: 10.1109/cvpr.2009.5206848.

[21] *DensePose*. URL: http://densepose.org/ (visited on 02/04/2021).

[22] Terrance DeVries and Graham W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. Nov. 29, 2017. arXiv: 1708.04552 [cs]. URL: http://arxiv.org/abs/1708.04552 (visited on 08/07/2021).

[23] Santosh K. Divvala, Ali Farhadi, and Carlos Guestrin. "Learning Everything about Anything: Webly-Supervised Visual Concept Learning." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 3270–3277. DOI: 10.1109/CVPR.2014.412.

[24] Vincent Dumoulin and Francesco Visin. *A Guide to Convolution Arithmetic for Deep Learning*. Jan. 11, 2018. arXiv: 1603.07285 [cs, stat]. URL: http://arxiv.org/abs/1603.07285 (visited on 02/14/2021).

[25] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge." In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-009-0275-4.

[26] Christoph Feichtenhofer et al. "SlowFast Networks for Video Recognition." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 6201–6210. DOI: 10.1109/ICCV.2019.00630.

[27] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. "Progressive Search Space Reduction for Human Pose Estimation." In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, AK, USA: IEEE, June 2008, pp. 1–8. ISBN: 978-1-4244-2242-5. DOI: 10.1109/cvpr.2008.4587468.

[28] Chelsea Finn et al. *Learning Visual Feature Spaces for Robotic Manipulation with Deep Spatial Autoencoders*. 2015. URL: http://www.icsi.berkeley.edu/pubs/vision/learningvisualfeaturespaces15.pdf.

[29] Yanwei Fu et al. "Interestingness Prediction by Robust Learning to Rank." In: *Computer Vision – ECCV 2014*. Vol. 8690. Cham: Springer International Publishing, 2014, pp. 488–503. ISBN: 978-3-319-10604-5 978-3-319-10605-2. DOI: 10.1007/978-3-319-10605-2_32.

[30] R. Girshick. "Fast R-CNN." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[31] R. Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81.

[32] Daniel Groos, Heri Ramampiaro, and Espen AF Ihlen. "EfficientPose: Scalable Single-Person Pose Estimation." In: *Applied Intelligence* (Nov. 6, 2020). ISSN: 0924-669X, 1573-7497. DOI: 10.1007/s10489-020-01918-7.

[33] K. He et al. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[34] Kaiming He et al. "Mask R-CNN." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Feb. 1, 2020), pp. 386–397. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2018.2844175.

[35] Nikolas Hesse et al. "Estimating Body Pose of Infants in Depth Images Using Random Ferns." In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. Santiago, Chile: IEEE, Dec. 2015, pp. 427–435. ISBN: 978-1-4673-9711-7. DOI: 10.1109/iccvw.2015.63.

[36] Nikolas Hesse et al. "Computer Vision for Medical Infant Motion Analysis: State of the Art and RGB-D Data Set." In: *Computer Vision – ECCV 2018 Workshops*. Vol. 11134. Cham: Springer International Publishing, 2019, pp. 32–49. ISBN: 978-3-030-11023-9 978-3-030-11024-6. DOI: 10.1007/978-3-030-11024-6_3.

[37] Zhongxu Hu et al. "Deep Convolutional Neural Network-Based Bernoulli Heatmap for Head Pose Estimation." In: *Neurocomputing* (Jan. 2021), S0925231221000692. ISSN: 09252312. DOI: 10.1016/j.neucom.2021.01.048.

[38] Donald F. Huelke. "An Overview of Anatomical Considerations of Infants and Children in the Adult World of Automobile Safety Design." In: *Annual Proceedings / Association for the Advancement of Automotive Medicine* 42 (1998), pp. 93–113. ISSN: 1540-0360. pmid: null. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3400202/ (visited on 01/08/2021).

[39] Omer F. Ince et al. "Child and Adult Classification Using Ratio of Head and Body Heights in Images." In: *International Journal of Computer and Communication Engineering* 3.2 (2014), pp. 120–122. ISSN: 20103743. DOI: 10.7763/ijcce.2014.v3.304.

[40] Omer Faruk Ince et al. *Child and Adult Classification Using Biometric Features Based on Video Analytics*. 2017.

[41] Eldar Insafutdinov et al. *DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model*. Nov. 30, 2016. arXiv: 1605.03170 [cs]. URL: http://arxiv.org/abs/1605.03170 (visited on 02/14/2021).

[42] Hueihan Jhuang et al. "Towards Understanding Action Recognition." In: *2013 IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, Dec. 2013, pp. 3192–3199. ISBN: 978-1-4799-2840-8. DOI: 10.1109/iccv.2013.396.

[43] Sheng Jin et al. *Towards Multi-Person Pose Tracking: Bottom-up and Top-down Methods*. 2017.

[44] Sam Johnson and Mark Everingham. "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation." In: *Procedings of the British Machine Vision Conference 2010*. Aberystwyth: British Machine Vision Association, 2010, pp. 12.1–12.11. ISBN: 978-1-901725-40-7. DOI: 10.5244/c.24.12.

[45] Sam Johnson and Mark Everingham. "Learning Effective Human Pose Estimation from Inaccurate Annotation." In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, June 2011, pp. 1465–1472. ISBN: 978-1-4577-0394-2. DOI: 10.1109/cvpr.2011.5995318.

[46] Will Kay et al. *The Kinetics Human Action Video Dataset*. May 19, 2017. arXiv: 1705.06950 [cs]. URL: http://arxiv.org/abs/1705.06950 (visited on 12/03/2020).

[47] Alvin Kim and Juan Manuel Camacho. *A Point Feature Matching-Based Approach To Real-Time Camera Video Stabilization*. 2018.

[48] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. "VIBE: Video Inference for Human Body Pose and Shape Estimation." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 5252–5262. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00530.

[49] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. May 8, 2012.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[51] H. Kuehne et al. "HMDB: A Large Video Database for Human Motion Recognition." In: *2011 International Conference on Computer Vision*. Barcelona, Spain: IEEE, Nov. 2011, pp. 2556–2563. ISBN: 978-1-4577-1102-2 978-1-4577-1101-5 978-1-4577-1100-8. DOI: 10.1109/iccv.2011.6126543.

[52] Y. LeCun, Fu Jie Huang, and L. Bottou. "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting." In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. June 2004, II–104 Vol.2. DOI: 10.1109/cvpr.2004.1315150.

[53] Jia Li, Wen Su, and Zengfu Wang. *Simple Pose: Rethinking and Improving a Bottom-up Approach for Multi-Person Pose Estimation*. Nov. 24, 2019. arXiv: 1911.10529 [cs]. URL: http://arxiv.org/abs/1911.10529 (visited on 02/08/2021).

[54] Panfeng Li, Youzuo Lin, and Emily Schultz-Fellenz. *Contextual Hourglass Network for Semantic Segmentation of High Resolution Aerial Imagery*. Feb. 9, 2019. arXiv: 1810.12813 [cs]. URL: http://arxiv.org/abs/1810.12813 (visited on 02/09/2021).

[55] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context." In: *arXiv e-prints*, arXiv:1405.0312 (May 2014), arXiv:1405.0312. arXiv: 1405.0312 [cs.CV].

[56] Huajun Liu et al. *Polarized Self-Attention: Towards High-Quality Pixel-Wise Regression*. July 8, 2021. arXiv: 2107.00782 [cs]. URL: http://arxiv.org/abs/2107.00782 (visited on 08/05/2021).

[57] Y. Luo et al. "LSTM Pose Machines." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 5207–5215. DOI: 10.1109/CVPR.2018.00546.

[58] Zhengxiong Luo et al. *Rethinking the Heatmap Regression for Bottom-up Human Pose Estimation*. Jan. 5, 2021. arXiv: 2012.15175 [cs]. URL: http://arxiv.org/abs/2012.15175 (visited on 01/06/2021).

[59] Diogo C. Luvizon, Hedi Tabia, and David Picard. "Human Pose Regression by Combining Indirect Part Detection and Contextual Information." In: *Computers & Graphics* 85 (Dec. 1, 2019), pp. 15–22. ISSN: 0097-8493. DOI: 10.1016/j.cag.2019.09.002.

[60] L. V. D. Maaten and Geoffrey E. Hinton. "Visualizing Data Using T-SNE." In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[61] William McNally et al. *EvoPose2D: Pushing the Boundaries of 2D Human Pose Estimation Using Neuroevolution*. Nov. 17, 2020. arXiv: 2011.08446 [cs]. URL: http://arxiv.org/abs/2011.08446 (visited on 08/05/2021).

[62] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. "PoseFix: Model-Agnostic General Human Pose Refinement Network." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 7765–7773. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00796.

[63] *MPII Leader Board*. URL: http://human-pose.mpi-inf.mpg.de/#results.

[64] Tewodros Legesse Munea et al. "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation." In: *IEEE Access* (2020). DOI: 10.1109/access.2020.3010248.

[65] Nurliyana Muty and Zati Azizul Hasan. "Detecting Arm Flapping in Children with Autism Spectrum Disorder Using Human Pose Estimation and Skeletal Representation Algorithms." In: Aug. 1, 2016, p. 6. DOI: 10.1109/ICAICTA.2016.7803118.

[66] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked Hourglass Networks for Human Pose Estimation." In: *Computer Vision – ECCV 2016*. Vol. 9912. Cham: Springer International Publishing, 2016, pp. 483–499. ISBN: 978-3-319-46483-1 978-3-319-46484-8. DOI: 10.1007/978-3-319-46484-8_29.

[67] Aiden Nibali et al. *Numerical Coordinate Regression with Convolutional Neural Networks*. May 3, 2018. arXiv: 1801.07372 [cs]. URL: http://arxiv.org/abs/1801.07372 (visited on 01/28/2021).

[68] Teo T. Niemirepo, Marko Viitanen, and Jarno Vanne. "Binocular Multi-CNN System for Real-Time 3D Pose Estimation." In: *Proceedings of the 28th ACM International Conference on Multimedia*. Seattle WA USA: ACM, Oct. 12, 2020, pp. 4553–4555. ISBN: 978-1-4503-7988-5. DOI: 10.1145/3394171.3414456.

[69] Zhenxing Niu et al. "Ordinal Regression with Multiple Output CNN for Age Estimation." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 4920–4928. ISBN: 978-1-4673-8851-1. DOI: 10.1109/cvpr.2016.532.

[70] Augustus Odena, Vincent Dumoulin, and Chris Olah. "Deconvolution and Checkerboard Artifacts." In: *Distill* 1.10 (Oct. 17, 2016), e3. ISSN: 2476-0757. DOI: 10.23915/distill.00003.

[71] Feyisayo Olalere. "Video-Based Activity Recognition for Child Behaviour Understanding." Msc Thesis. Utrecht, Netherlands: Utrecht University, July 2021.

[72] G. Papandreou et al. "Towards Accurate Multi-Person Pose Estimation in the Wild." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 3711–3719. DOI: 10.1109/CVPR.2017.395.

[73] George Papandreou et al. *PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model*. Mar. 22, 2018. arXiv: 1803.08225 [cs]. URL: http://arxiv.org/abs/1803.08225 (visited on 02/13/2021).

[74] Tomas Pfister et al. "Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos." In: *Computer Vision – ACCV 2014*. Vol. 9003. Cham: Springer International Publishing, 2015, pp. 538–552. ISBN: 978-3-319-16864-7 978-3-319-16865-4. DOI: 10.1007/978-3-319-16865-4_35.

[75] Leonid Pishchulin et al. *DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation*. Apr. 26, 2016. arXiv: 1511.06645 [cs]. URL: http://arxiv.org/abs/1511.06645 (visited on 02/08/2021).

[76] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. Feb. 26, 2021. arXiv: 2103.00020 [cs]. URL: http://arxiv.org/abs/2103.00020 (visited on 05/05/2021).

[77] Shyam Sundar Rajagopalan, Abhinav Dhall, and Roland Goecke. "Self-Stimulatory Behaviours in the Wild for Autism Diagnosis." In: *2013 IEEE International Conference on Computer Vision Workshops*. Sydney, Australia: IEEE, Dec. 2013, pp. 755–761. ISBN: 978-1-4799-3022-7. DOI: 10.1109/ICCVW.2013.103.

[78] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. Apr. 8, 2018. arXiv: 1804.02767 [cs]. URL: http://arxiv.org/abs/1804.02767 (visited on 05/04/2021).

[79] S. Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1137–1149. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2016.2577031.

[80] K. Ricanek and T. Tesafaye. "MORPH: A Longitudinal Image Database of Normal Adult Age-Progression." In: *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. Southampton, UK: IEEE, 2006, pp. 341–345. ISBN: 978-0-7695-2503-7. DOI: 10.1109/fgr.2006.78.

[81] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. May 18, 2015. arXiv: 1505.04597 [cs]. URL: http://arxiv.org/abs/1505.04597 (visited on 02/13/2021).

[82] Ben Sapp and Ben Taskar. "MODEC: Multimodal Decomposable Models for Human Pose Estimation." In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, June 2013, pp. 3674–3681. ISBN: 978-0-7695-4989-7. DOI: 10.1109/cvpr.2013.471.

[83] Giuseppa Sciortino et al. "On the Estimation of Children's Poses." In: *Image Analysis and Processing - ICIAP 2017*. Cham: Springer International Publishing, 2017, pp. 410–421. ISBN: 978-3-319-68548-9.

[84] Shenghao Shi. *Facial Keypoints Detection*. Oct. 15, 2017. arXiv: 1710.05279 [cs, stat]. URL: http://arxiv.org/abs/1710.05279 (visited on 01/28/2021).

[85] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Apr. 10, 2015. arXiv: 1409.1556 [cs]. URL: http://arxiv.org/abs/1409.1556 (visited on 02/08/2021).

[86] Ke Sun et al. "Deep High-Resolution Representation Learning for Human Pose Estimation." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 5686–5696. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00584.

[87] Xiao Sun et al. "Integral Human Pose Regression." In: *Computer Vision – ECCV 2018*. Vol. 11210. Cham: Springer International Publishing, 2018, pp. 536–553. ISBN: 978-3-030-01230-4 978-3-030-01231-1. DOI: 10.1007/978-3-030-01231-1_33.

[88] Jonathan Tompson et al. *Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation*. Sept. 17, 2014. arXiv: 1406.2984 [cs]. URL: http://arxiv.org/abs/1406.2984 (visited on 12/04/2020).

[89] Alexander Toshev and Christian Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), pp. 1653–1660. DOI: 10.1109/CVPR.2014.214. arXiv: 1312.4659.

[90] w3techs.com. *Usage Statistics and Market Share of Content Languages for Websites, August 2021*. URL: https://w3techs.com/technologies/overview/content_language (visited on 08/02/2021).

[91] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. "An Approach to Pose-Based Action Recognition." In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, June 2013, pp. 915–922. ISBN: 978-0-7695-4989-7. DOI: 10.1109/CVPR.2013.123.

[92] Jingdong Wang et al. *Deep High-Resolution Representation Learning for Visual Recognition*. Mar. 13, 2020. arXiv: 1908.07919 [cs]. URL: http://arxiv.org/abs/1908.07919 (visited on 02/06/2021).

[93] S. Wei et al. "Convolutional Pose Machines." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 4724–4732. DOI: 10.1109/CVPR.2016.511.

[94] Bin Xiao, Haiping Wu, and Yichen Wei. *Simple Baselines for Human Pose Estimation and Tracking*. Aug. 21, 2018. arXiv: 1804.06208 [cs]. URL: http://arxiv.org/abs/1804.06208 (visited on 02/06/2021).

[95] Sen Yang, Wankou Yang, and Zhen Cui. *Pose Neural Fabrics Search*. Dec. 5, 2020. arXiv: 1909.07068 [cs]. URL: http://arxiv.org/abs/1909.07068 (visited on 08/05/2021).

[96] Feng Zhang et al. "Distribution-Aware Coordinate Representation for Human Pose Estimation." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 7091–7100. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00712.

[97] Hang Zhang et al. *ResNeSt: Split-Attention Networks*. Dec. 30, 2020. arXiv: 2004.08955 [cs]. URL: http://arxiv.org/abs/2004.08955 (visited on 02/09/2021).

[98] Hong Zhang et al. *Human Pose Estimation with Spatial Contextual Information*. Jan. 7, 2019. arXiv: 1901.01760 [cs]. URL: http://arxiv.org/abs/1901.01760 (visited on 01/26/2021).

[99] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. "From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding." In: *2013 IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, Dec. 2013, pp. 2248–2255. ISBN: 978-1-4799-2840-8. DOI: 10.1109/ICCV.2013.280.

[100] Zhifei Zhang, Yang Song, and Hairong Qi. "Age Progression/Regression by Conditional Adversarial Autoencoder." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 4352–4360. ISBN: 978-1-5386-0457-1. DOI: 10.1109/cvpr.2017.463.

Appendices

Appendix A

Actions in *Kinetikids*

A.1 Action Categories

Bowling
Baseball
Basketball
Softball
Kickball
Golfing
Cricket
Tennis
Soccer
Volleyball
American football
Parasailing
Surfing water
Water skiing
Windsurfing
Hurdling
Parkour
Archery
Frisbee
Disc golfing
Throwing ball
Throwing discus
Badminton
Bouncing on trampoline
Cartwheeling
Gymnastics tumbling
Somersaulting
Gymnastics Vault

A.2 Action Labels

doing archery
bouncing on trampoline
bowling
cartwheeling
catching baseball
throwing baseball
catching frisbee
throwing frisbee
catching softball
throwing softball
disc golfing
dribbling basketball
dunking basketball
golf chipping
golf driving
golf putting
doing gymnastic tumbling
hitting baseball
hurdling
juggling soccer ball
kicking field goal
kicking soccer ball
parasailing
doing parkour
passing American football
passing American football
playing badminton
playing basketball
playing cricket
playing kickball
playing tennis
playing volleyball
shooting basketball
shooting goal (soccer)
somersaulting
surfing water
throwing ball
throwing discus
doing gymnastics vault
doing water skiing
doing windsurfing

Appendix B

Annotation Information

B.1 Keypoints

- Nose Tip
- Left Ear
- Right Ear
- Left Eye
- Right Eye
- Left Shoulder
- Right Shoulder
- Left Elbow
- Right Elbow
- Left Wrist
- Right Wrist
- Left Hip
- Right Hip
- Left Knee
- Right Knee
- Left Ankle
- Right Ankle

Appendix C

Queries - Google Images

C.1 Query templates

kids

childs

toddlers

middle school

preschool

primary school

elementary school

C.2 Query templates

<subject> *<action label>*

<subject> *<action label>* competition

<action label> with *<subject>*

C.3 Languages

English - en

Russian - ru

Turkish - tr

Spanish - es

Persian - fa

French - fr

German - de

Japanese - ja

Vietnamese - vi

Chinese (simplified) - zh-cn

Chinese (traditional) - zh-tw

Arabian - ar

Portuguese - pt

Greek - el

Italian - it

Indonesian - id

Ukranian - uk

Polish - pl

Dutch - nl

Korean - ko

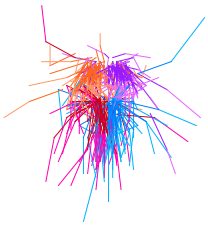
Hebrew - iw

Appendix D

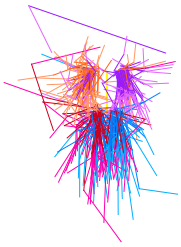
Dataset Visualizations

Figure D.1: Example images with annotations from each of the compiled datasets. From top to bottom, we visualize poses from COCO Adult, COCO Child, and *Kinetikids-pose*.

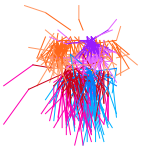
(a) COCO Adult (Frontal)



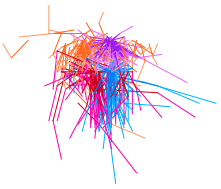
(b) COCO Child (Frontal)



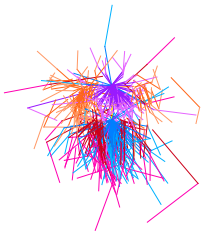
(c) Kinetikids (Frontal)



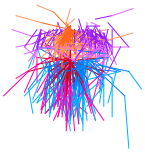
(d) COCO Adult (Left)



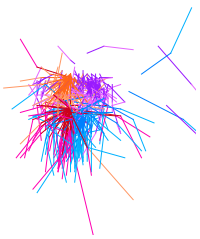
(e) COCO Child (Left)



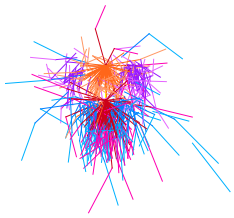
(f) Kinetikids (Left)



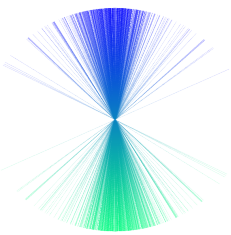
(g) COCO Adult (Right)



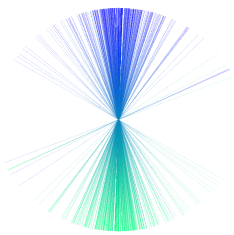
(h) COCO Child (Right)



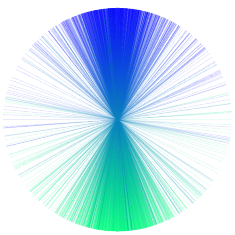
(i) Kinetikids (Right)



(j) COCO Adult (Right)



(k) COCO Child (Right)



(l) Kinetikids (Right)

Figure D.2: Poses of datasets