Human Pose Estimation: Progress Report

Robert Lee, Julian Rocha, Wanze Zhang, Nicole Peverley, Rafay Chaudhry, Corey Koelewyn

March 14, 2021

1 The Problem

Human pose estimation (HPE) is the problem domain of identifying body keypoints to construct a body model. Many existing systems accept images as input, with some implementations accepting data such as point cloud or videos. The application of HPE is widespread and benefits many industries. In particular, HPE can revolutionize industries such as augmented reality, animation, gaming, and robotics [1]. There are examples where a person's gait can be used as an unique fingerprint for tracking or identifying individuals in videos [2]. Another subset of HPE is hand pose estimation which can be used to translate sign language. HPE is a difficult problem domain due to challenges such as variability in human appearance and physique, environment lighting and weather, occlusions from other objects, self-occlusions from overlapping joints, complexity of movements of the human skeleton, and the inherent loss of information with a 2D image input [3]. This largely unsolved problem enables us to explore many novel and creative approaches, enriching our learning experience. We are excited to explore these applications, but we decided to limit our scope to a general version of the problem so we could reference the abundance of research available.

There are many variations of HPE systems, which can be roughly categorized into 2D vs 3D, single-person vs multi-person, and different body models. Our group plans to focus on single-frame monocular RGB images containing a single individual rather than a photo with multiple individuals. We believe it will be more feasible to train a deep neural network (NN) with one individual based upon the research papers available and in the given timeframe. Given success with single individuals, we may explore multi-person HPE. Current state-of-the-art techniques for 2D single-person HPE can be categorized into two categories: regression on absolute joint position, or detection on joint locations with heat maps. Since a direct mapping from the input space to joint coordinates is a highly non-linear problem, heat-map-based approaches have proven to be more robust by including small-region information [4]. Thus, we have used the heat map approach.

There are three different types of models used with full body HPE: kinematic, contour, and volumetric, as shown in Fig. 1 [4]. A kinematic model consists of points on each human joint connected by straight lines, similar to a stick-figure skeleton. The contour model consists of 2D squares and rectangles that represent the body, and the volumetric model represents the body with 3D cylinders. Further high-fidelity models implement meshes that capture more details of the human pose. The kinematic model is the simplest model to perform loss metric computations, and thus is preferred by our group as a scope-limiting decision. Using a basic kinematic model will

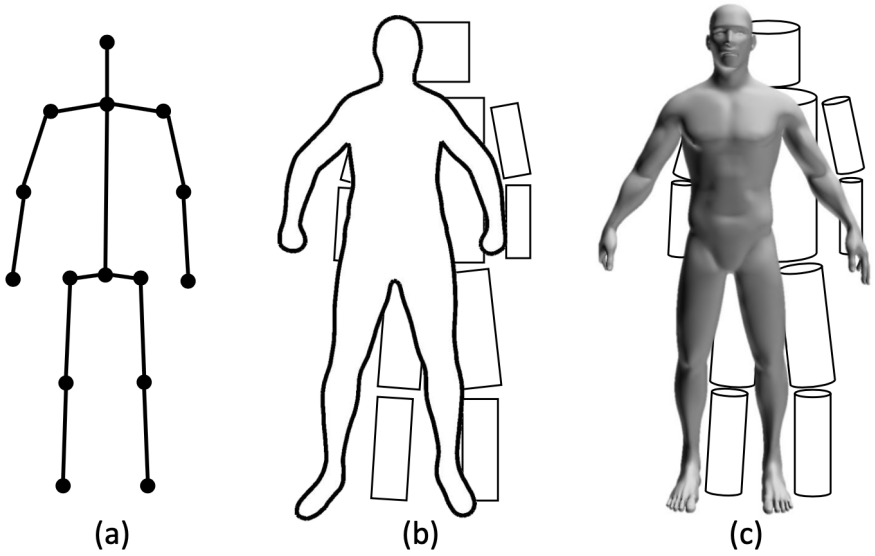


Figure 1: Common body models: (a) skeleton-based, (b) contour-based, (c) volume-based [4]

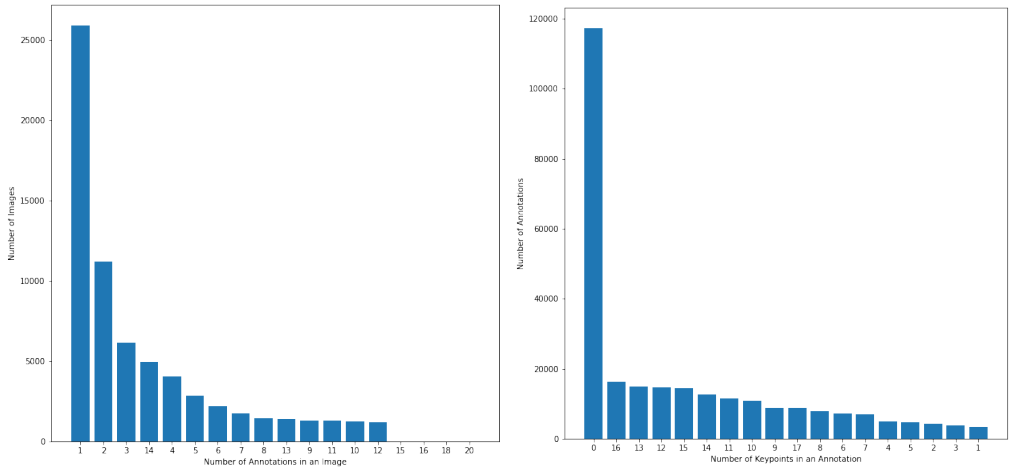
simplify the problem space and encourage us to focus on tangible results that can be used in real life applications. Our goal is to estimate a kinematic model for the individual in each picture.

While there were many existing HPE datasets, very few perfectly matched our chosen requirements for the project. Any available datasets would need to be cleaned and processed. Many datasets contained assorted images that used different types of models for the pose estimation or consisted of multiple people in each image. Our primary choice is the COCO dataset [5]. This dataset consists of 330 K images, of which 200 K are labelled. There are pre-sorted subsets of this dataset specific for HPE competitions: COCO16 and COCO17. These contain 147 K images labelled with bounding boxes, joint locations, and human body masks. We decided to use COCO17 for our generator because. We have decided against using DensePose [6], which is a highly detailed manually annotated subset of the COCO dataset, because it does not offer joint coordinate labels. We hope to use the MPII dataset [7], which consists of 41 K labelled images split into 29 K train and 12 K test, for validating our model's performance if time permits.

2 Goals

3 Plans and Progress To Date

Various metrics were gathered on the COCO dataset to help inform how the data should be processed as well as how the model should handle different scenarios. The two metrics most impactful to the project thus far are documented in Fig. 2. There are 66'808 images in the dataset containing a total of 273'469 annotations. Despite the fact that we have chosen to tackle single person and not multi person pose estimation, a large number of the COCO images contain more than one annotated person. It would be desirable if we did not have to discard these images, so cropping to



(a) Number of images with a given number of annotations

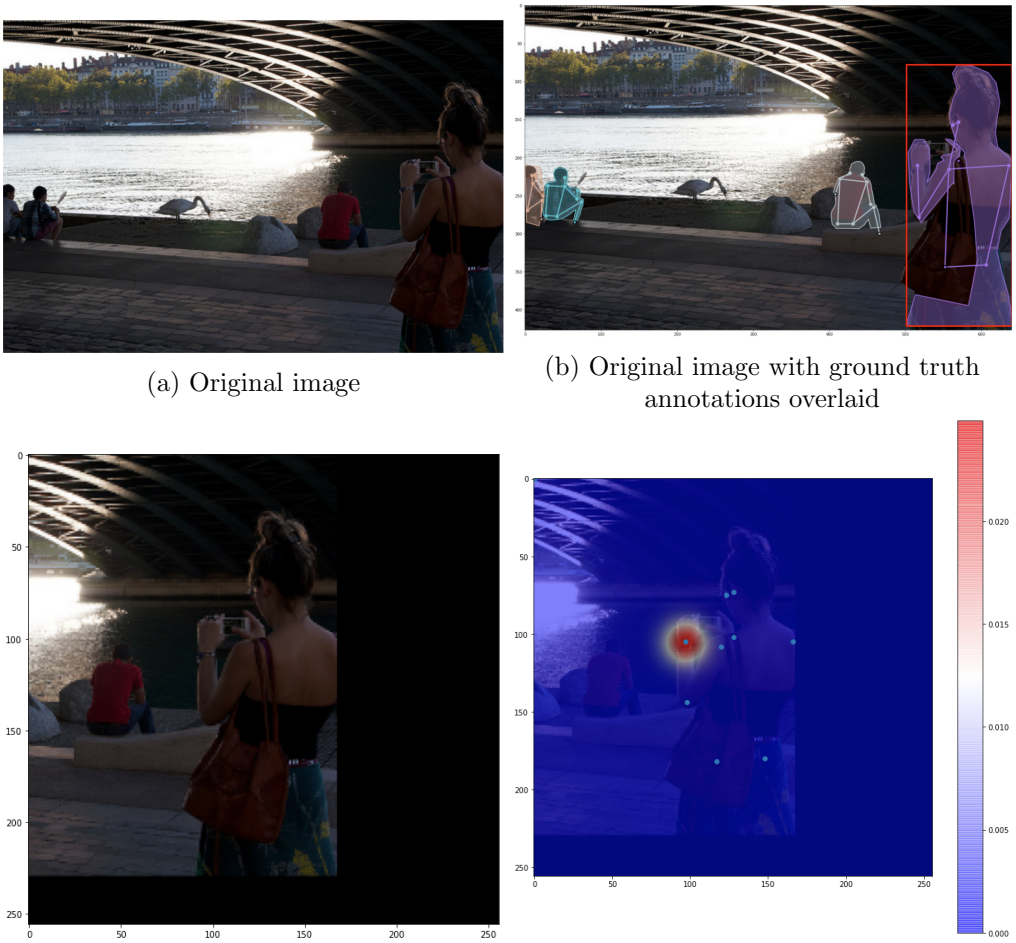
(b) Number of annotations with a given number of keypoints

Figure 2: COCO metrics

Figure 3: Examples of annotations with zero labelled keypoints

a bounding box can allow us to convert a multi person image into a single person image. Most of the annotations in the dataset do not have all 17 key points of the body labelled. The model should not expect a perfect human image with all key points visible in frame. The model should instead output a dynamic number of keypoints based on what it can find. But what is the purpose of an annotation with 0 labelled keypoints? Fig. 3 shows two examples of these 0 keypoint annotations. Clearly the bounding boxes of these annotations denote people, but because there are no labeled keypoints, these examples may confuse our model. Therefore, despite the fact that 0 keypoint annotations make up 42.89% of the total dataset annotations, these annotations will be filtered out during training.

The COCO dataset is more than 20GB so keeping the entire dataset in memory during training is not an option. Image preprocessing needs to be done to get the images and annotations in a format that can be passed to the model. A data generator was developed to tackle these two tasks. To prevent the generator from being the bottleneck of the training process, it makes use of multiple cores. The generator fetches images from disk in batches and the next batch can be fetched and processed while the model is performing the front and back propagation on the current batch.



(a) Original image

(b) Original image with ground truth annotations overlaid

(c) Processed image

(d) Processed image with translated keypoints and left hand heat map overlaid

Figure 4: Example of transformations applied by the data generator

The preprocessing responsibilities of the data generator include: cropping to the ground truth bounding box of a person, resizing to the models input resolution and dimensions, and converting ground truth annotations for each keypoint to a heatmap for the cropped images. Fig. 4 shows an example of the transformations. Fig. 4 only shows the cropping for one person but since there are 4 annotated people, the image would get split into 4 images, each centered on the person of interest. Fig. 4 also only shows the heat map of the left hand, but since COCO annotations contain 17 keypoints, it produces 17 heatmaps per annotation.

4 Task Breakdown

Julian is planning on performing experiments to see how adding data augmentation to the input pipeline will affect model performance. These experiments will include: varying bounding box size from 110% to 150%, flipping images horizontally, rotating images slightly, and various adjustments to brightness, contrast, and noise/grain. Augmentation will be done online between batch fetches

and will amplify rather than replace examples. Due to the long training times of the current model, Julian may explore the option of converting a keras model to a TPU compatible model, which can be run on Google Collab TPU's to hopefully reduce training time. Finally, Julian would like to explore hyperparameter tuning and training of the model.

5 Initial Results

References

[1] Pose Estimation Guide — Fritz AI. [Online]. Available: https://www.fritz.ai/pose-estimation/

[2] W. Zeng and C. Wang, "Human gait recognition via deterministic learning," *Neural Networks*, vol. 35, pp. 92–102, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S089360801200192X

[3] L. Sigal, *Human Pose Estimation*. Boston, MA: Springer US, 2014, pp. 362–370. [Online]. Available: https://doi.org/10.1007/978-0-387-31439-6_584

[4] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," *Computer Vision and Image Understanding*, vol. 192, p. 102897, Mar 2020. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2019.102897

[5] Common Objects in Context. [Online]. Available: https://cocodataset.org/#home

[6] DensePose. [Online]. Available: http://densepose.org/

[7] MPII Human Pose Database. [Online]. Available: http://human-pose.mpi-inf.mpg.de/#dataset