Human Pose Estimation

Madhava Palyiam
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
mpaliyam@terpmail.umd.edu

Vatsal Agarwal
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
vatsalag99@gmail.com

Helen Chen
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
hchen104@umd.edu

Nikhil Pateel
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
npateel@umd.edu

Abstract

The detection of human joints (e.g. wrists, knees, elbows, etc.) in images and videos has always been a difficult task. By using deep neural networks (DNN), this can aid in being able to detect a human body in still pictures and estimating precisely where those specific human joints are located. We will discuss different types of methods for human pose estimation and some of the challenges that are faced with these methods.

1 Introduction

Human poses are important in many different technological environments. In the field of Augmented Reality (AR) and Computer-generated Imagery (CGI), poses are used to generate and display realistic and natural body movements [Kumarapu and Mukherjee, 2020], while in national security, cameras that use capture poses can assess the movement of people in public places, and can even be used to flag individuals as suspicious [Mathis and Mathis, 2020]. In medicine, human poses can be used to allow practitioners to monitor a patient's movements and help make diagnoses related to posture or gait [Chen et al., 2018]. In order to get the benefits of any of these technologies however, human pose information must be extracted from images or video. However, human pose estimation is a difficult problem in the field of computer vision. The human body's small joints are particularly difficult for computers to recognize, and therefore a variety of techniques and algorithms are needed to capture them. In this paper, we shall be examining several such recent approaches on estimating poses.

2 Different Practices of Human Pose Estimation

There are two main approaches to multi-person pose estimation. The first uses a top down approach, where an the algorithm first detects individual people and then individually runs a pose detection algorithm for each recognized person, while the second approach is a bottom-up, Here, key points or "joints" are first detected throughout the entire image, and then joints and connections are assigned to different people.

2.1 Top-Down Approach

As mentioned earlier, a top-down approach to pose-detection aims at first identifying individual people using object detection methods and then runs a separate algorithm to determine the poses for a single

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

person. However, this can be problematic since the performance of the pose estimation is very much dependent on the quality of the human bounding boxes. Thus recently, a regional multi-person pose estimation framework named AlphaPose has been proposed to limit the effect of these inaccuracies. The main problems that the authors identify with human detection algorithms is the issue of bounding box localization errors and redundant detections. The former results in important parts of a person being missed by the bounding box causing the pose estimation to be incorrect, while the latter can result in redundant detections since a pose is generated for each bounding box [Fang et al., 2016].

Their proposed method consists of three modules: Symmetric Spatial Transformer Network (SSTN), Parametric Pose Non-Maximum-Suppression (NMS) and Pose-Guided Proposals Generator (PGPG). The SSTN module is combined with a standard off-the-shelf single-person pose estimator network (SPPE) to take in the bounding boxes and generate the possible poses. These are then fed to the custom NMS algorithm (p-Pose NMS) to obtain more refined proposals by removing repetitive poses. They additionally apply the pose-guided proposals generator to increase the number of training examples [Fang et al., 2016].

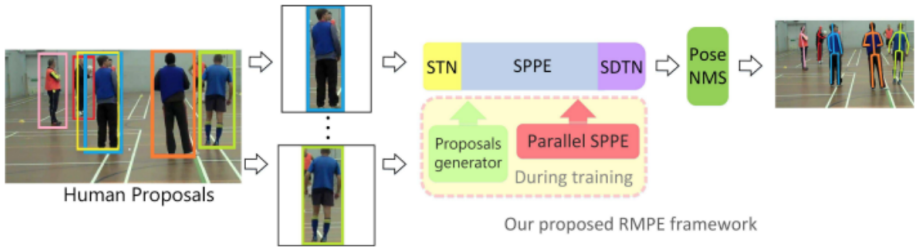

Figure 1: AlphaPose model structure.

With regards to the pose generation, their SSTN is composed of a spatial transformer network that is able to automatically capture regions of interest such as human poses and a spatial de-transformer which acts as a transformation matrix to map the pose key points back to the original image coordinates [Jaderberg et al., 2015]. These modules are added to the beginning and end of an SPPE network and then further augmented by a parallel SPPE that only has the initial STN module. This parallel branch aims to restrict the STN from choosing poses that are not located in the center, and thus forces it to focus on the important regions in the image. It is important to note that this branch is only active during the training and is then unused in testing. For the non-max-suppression, the authors define a pose distance metric to determine whether a pose should be eliminated for being too close to another one. This metric is composed of two functions, one for matching points based on confidence and one for matching points based on distance. The parameters passed into this model are then optimized two at a time to aid in better performance [Fang et al., 2016].

To ensure the robustness of their pose estimation pipeline, the authors apply the PGPG algorithm as data augmentation. They do this by approximating the distribution of the offset between the predicted and ground truth bounding boxes given the individual pose. Given the difficulty to learn this distribution due to natural variations in each human pose, the authors compute the atomic pose [Yao and Fei-Fei, 2012] of P and condition the distribution on that. Thus, the authors align all of the poses and then apply clustering to obtain the centroids of the different groups, which are treated as the atomic pose. Lastly, they calculate the offsets between the ground truth and predicted bounding box for each unique atomic pose and fit the normalized data to a Gaussian and then sample from it to obtain new offsets to apply and augment their data [Fang et al., 2016].

In the paper, the authors use a VGG-backbone SSD-512 network [Liu et al., 2015] to generate the bounding boxes and use the state-of-the-art stacked hourglass model [Newell et al., 2016] as the SPPE. For the SSTN module, they use the ResNet-18 [He et al., 2015] for the STN network and the 4-stack hourglass network as the parallel SPPE. Lastly, the authors evaluate their proposed methodology on two standard benchmarks, the MPII Multi-Person Dataset [Andriluka et al., 2014] and the MSCOCO Keypoints Challenge [Lin et al., 2014]. Overall, it is found that the SSTN and the parallel SPPE are especially important in performance as they ensure that the poses are high quality and chosen from the correct region. Furthermore, the introduced data augmentation using the Gaussian distribution is better than the standard baseline of random jittering. Lastly, it is shown that

the use of the custom NMS algorithm performs better and is faster than previous approaches which did not optimize parameters [Fang et al., 2016].

	Methods	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
	RMPE, full	**90.7**	**89.7**	**84.1**	**75.4**	**80.4**	**75.5**	**67.3**	**80.8**
a)	w/o SSTN+parallel SPPE	89.0	86.9	82.8	73.5	77.1	73.3	65.0	78.2
	w/o parallel SPPE only	89.9	88.0	83.4	74.7	77.8	74.0	65.8	79.1
b)	w/o PGPG	82.8	81.0	77.5	68.2	74.6	66.8	60.1	73.0
	random jittering*	89.3	87.8	82.3	70.4	78.4	73.3	63.8	77.9
	w/o PoseNMS	85.1	83.6	79.2	69.8	76.4	72.2	63.6	75.7
c)	PoseNMS (2015)	88.9	87.8	83.0	73.8	78.7	74.6	66.3	79.1
	PoseNMS (2013)	90.0	88.6	83.7	74.6	79.7	75.1	67.0	79.9
d)	straight forward two-steps	81.9	80.4	74.1	68.5	69.0	66.1	62.2	71.7
e)	oracle human detection	94.3	93.4	87.7	80.2	84.3	78.9	70.6	84.2

Table 1: (Taken from Fang et al., 2016). Table shows the various accuracy each algorithm has in recognizing various body parts

2.2 Bottom-up Approach

A bottom up approach consists of detecting key points on an image and then connecting them based based on which person the key points belong to Cao et al. [2018]. According to the authors, this is a NP-hard problem, so there have been some attempts at approximating it, such as in Insafutdinov et al. [2016] and Pishchulin et al. [2015]. DeeperCut works by first detecting all body part candidates which are a set of all the detections of body parts in an image and a set of body parts classes [Insafutdinov et al., 2016]. The detection of the body part candidates are done using a ResNet-152 architecture. Each body part candidate is assigned a score for each body part class . Insafutdinov et al. [2016] improves upon their previous method, DeepCut [Pishchulin et al., 2015], which had used multiple cost functions, clustering, and the branch and cut algorithm to determine if two body part candidates belong to the same person. DeeperCut adds a layer to the ResNet architecture which will predict relative position between joints as well. Some problems with this approach is that there can be limits with the number of poses that are detected due to the computation required for each image [Cao et al., 2018]. In addition, each image can take several minutes to compute poses for.

2.2.1 OpenPose

OpenPose claims to have developed a method for human pose detection which is faster than both the top-down and traditional bottom-up approaches, such as DeeperCut, by simultaneously detecting and assigning key points [Cao et al., 2018]. Their model consists of repeatedly refining the part affinity field predictions for pairs of key points and the predicted confidence maps over the image.

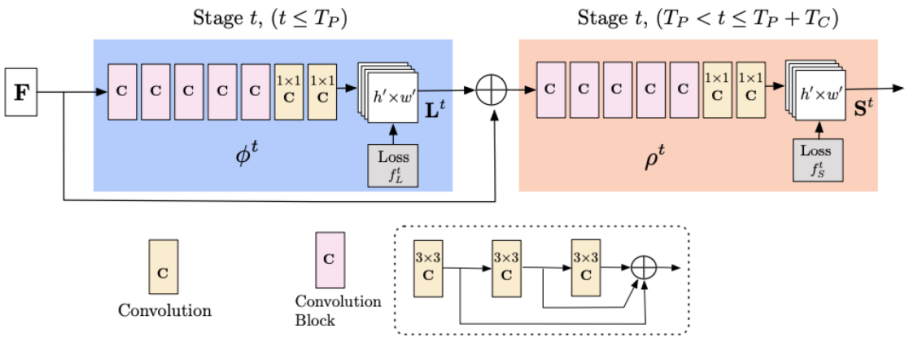

Figure 2: OpenPose Model Architecture

Their approach first uses a modified pertained VGG-19 model to generate a set of feature maps for each image. These feature maps are fed through the first part of their model which refines the part affinity field predictions. Their model has several convolutional blocks and 1x1 convolutional layers to generate part affinity fields which are fields which encode flow between predicted key points. They also concatenate predictions from the previous stage and the original feature vectors each time the input is fed through the model. Their model also creates a set of 2D confidence maps over the entire image, each corresponding to a particular key point such as right elbow or left shoulder, etc. A pixel in a particular map will have a high value if the particular limb that the map refers to appears on that pixel location. For example, if there were two people fully visible in the image then each of the maps would have two high values.

The next step is to assemble the point to assemble the human poses. The part affinity fields developed by the authors preserve the location and orientation information of the key point connections and this is very important to distinguish images with interactions between people. The confidence maps also encode information about location of key points and the types. They aggregate the confidence maps and part affinity fields in a method called bipartite matching and use the Hungarian Algorithm to find a graph which enforces constraints among the points such that no two limbs share the same part, while maximizing the affinity between them. The Hungarian Algorithm is a method of solving graph connections to maximize a reward, in this case correct poses [Cao et al., 2018]. In order to generate multiple graphs for different people, they use approximations to break the problem into multiple sub-problems and solve those using the Hungarian Algorithm [Cao et al., 2018]. OpenPose has also released an easily usable code library for implementing their method and running it on a personal computer. Their software can be run at 22 fps.

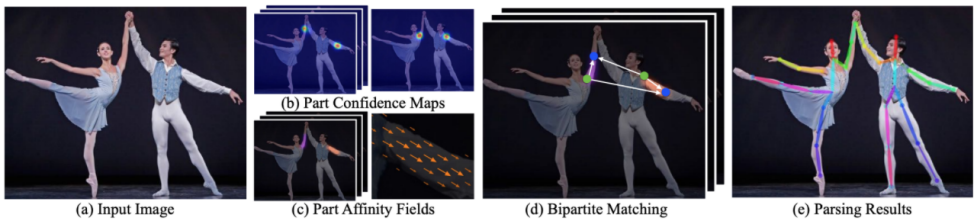

Figure 3: Steps in creating poses in OpenPose

3 Comparisons

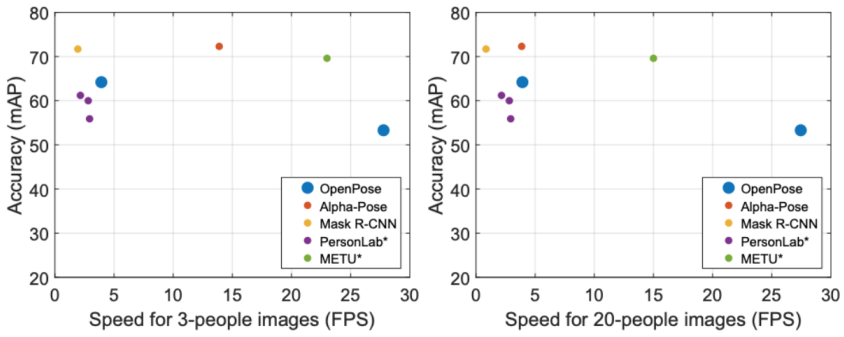

Figure 4: Comparisons of OpenPose with other algorithms, including He et al. [2017] and Fang et al. [2016]. Taken from Cao et al. [2018]

In the chart above, mAP (mean average precision) scores and frame speed are shown for some pose detection methods. In our report we covered two of these implementations in detail: OpenPose, and Alpha-Pose. Of these, OpenPose has the highest speed for both three person images and for 20 person images while retaining a modest mAP score. On the other hand if the frame speed is decreased,

OpenPose can achieve a higher mAP score. Alpha-Pose, a top-down approach performs with a high accuracy however it suffers from a slower frame rate. Finding a high frame rate is key to pose detection since a higher frame rate means closer to real time processing which is needed for many of the uses of pose detection. Therefore, developing methods that will retain high frame rate while also obtaining a high mAP score will most likely be the focus of future research and development in pose detection.

References

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. URL `http://arxiv.org/abs/1812.08008`.

K. Chen, P. Gabriel, A. Alasfour, C. Gong, W. K. Doyle, O. Devinsky, D. Friedman, P. Dugan, L. Melloni, T. Thesen, D. Gonda, S. Sattar, S. Wang, and V. Gilja. Patient-specific pose estimation in clinical environments. *IEEE Journal of Translational Engineering in Health and Medicine*, 6:1–11, 2018.

Haoshu Fang, Shuqin Xie, and Cewu Lu. RMPE: regional multi-person pose estimation. *CoRR*, abs/1612.00137, 2016. URL `http://arxiv.org/abs/1612.00137`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL `http://arxiv.org/abs/1703.06870`.

Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. *CoRR*, abs/1605.03170, 2016. URL `http://arxiv.org/abs/1605.03170`.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

Laxman Kumarapu and Prerana Mukherjee. Animepose: Multi-person 3d pose estimation and animation, 2020.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL `http://arxiv.org/abs/1405.0312`.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. URL `http://arxiv.org/abs/1512.02325`.

Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, 60:1–11, 2020.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. *CoRR*, abs/1511.06645, 2015. URL `http://arxiv.org/abs/1511.06645`.

B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703, 2012.