CSCE 5703: Computer Vision Project

Human Pose Estimation

Christy Dunlap
University of Arkansas
Department of Mechanical Engineering
cldunlap@uark.edu

Mishek Musa
University of Arkansas
Department of Mechanical Engineering
mjmusa@uark.edu

1. Introduction

1.1. What is human pose estimation?

Human pose estimation is a fundamental computer vision problem that involves estimating the spatial configuration of the human body in an image or video. It has significant applications in various fields, including robotics, sports analysis, healthcare, and entertainment. The task of accurately estimating human poses from images and videos is challenging due to the high degree of articulation and variability in human movements and appearances. The challenge of detecting poses becomes even more challenging when moving from a single-person estimation to multiple people. This is for a number of reasons including the detection of an unknown number of people that can appear at any pose or scale, the spatial inference issues that arise due to body contact between people or occlusions, and lastly, the runtime complexity grows proportionally with the number of people in the frame.

1.2. Approaches

Two key approaches have been proposed to solve the issues as it pertains to multi-person human pose estimation. The first is a top-down approach, whereby the model first detects people in the frame, and then independently estimates the pose of each person identified. This approach is able to make use of the existing single-person pose estimation algorithms, however, it suffers some key drawbacks. Firstly, if the person detector fails, there is no way for the model to recover and detect the pose of the unidentified people; this occurs mainly in scenarios where there are many occlusions and is known as the early commitment problem. Secondly, it does not solve the problem of the runtime complexity growing with the number of people. To overcome these issues, bottom-up approaches have been proposed. In this approach, the model first detects body parts or joints, and then a final parsing is done to extract the pose estimation result. This approach has the potential to overcome the early commitment problem as well as decouple the runtime from the number of people in the frame. For this final project, we review and implement one of the state-of-the-art bottom-up approaches to human pose estimation known as OpenPose [2]

1.3. Proposed Solution

OpenPose is a bottom-up approach for real-time multi-person pose estimation developed by researchers at Carnegie Mellon University and The University of California, Berkeley. It uses a deep neural network architecture to estimate the 2D locations of keypoints on the human body, such as the nose, shoulders, elbows, wrists, hips, knees, and ankles. Note that it is also capable of performing 3D pose estimation by combining the 3D estimated poses with camera calibration parameters. The novel contribution of this works includes the first bottom-up approach that uses association scores via part affinity fields which are essentially a set of 2D vector fields that encode the location and orientation of limbs over the image domain. This creates the sufficient global context for a greedy parse algorithm to achieve high-quality pose estimation results with minimal computational cost.

OpenPose was trained on three primary datasets, the MPII human multi-person dataset [1], the Common Objects in Context (COCO) dataset [3], and a custom COCO + foot dataset [2]. Each dataset is comprised of thousands of images with labeled human poses in a variety of complex articulations. Examples of the labeled images and the number of keypoints in each dataset can be seen in figure 1.

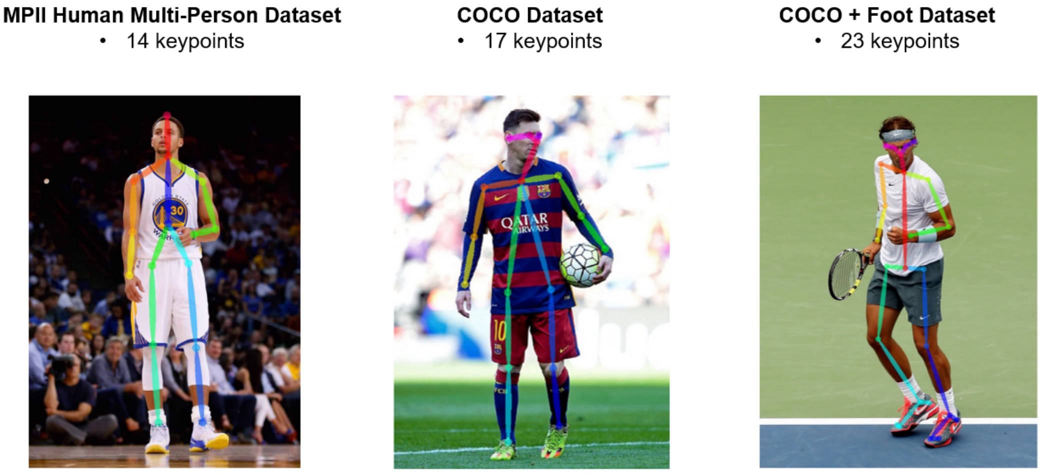


Figure 1. Examples from the three datasets used to train the OpenPose model.

2. Methods

2.1. Network Design

The general model predicts part affinity fields (PAF) and confidence maps [2]. A confidence map consists of values equal to the number of pixels in the image. Each value corresponds to a pixel and represents the confidence of that pixel being a specified joint. There is a confidence map for each joint. The part affinity fields describe the orientation and location of limbs. There is a PAF for each limb. Each PAF consists of a set of vectors equal to the number of pixels. Ideally if a pixel is on a limb then the vector will be a unit vector point parallel to the limb and zero otherwise. From these two outputs, they are able to determine the limb locations and piece them together for all people in the image.

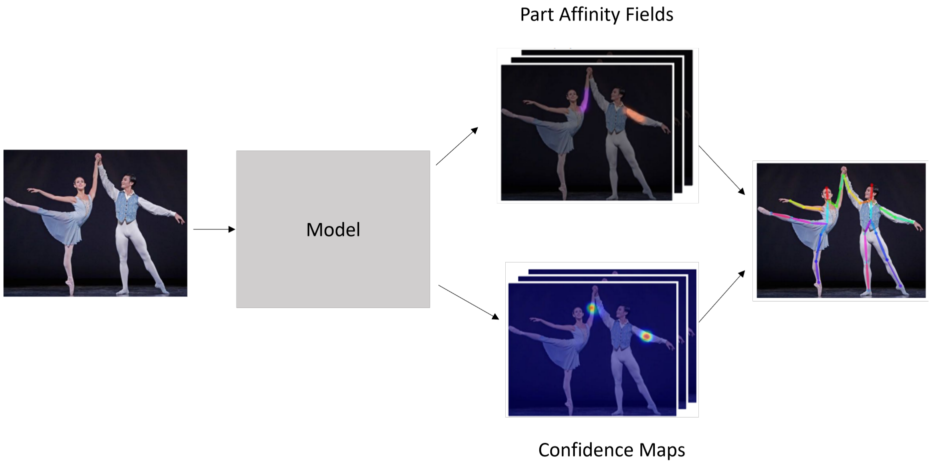


Figure 2. Overview of openpose model. The model takes in inputs of images and predicts PAF and confidence maps. Then these are used to approximate the pose of the people in the image.

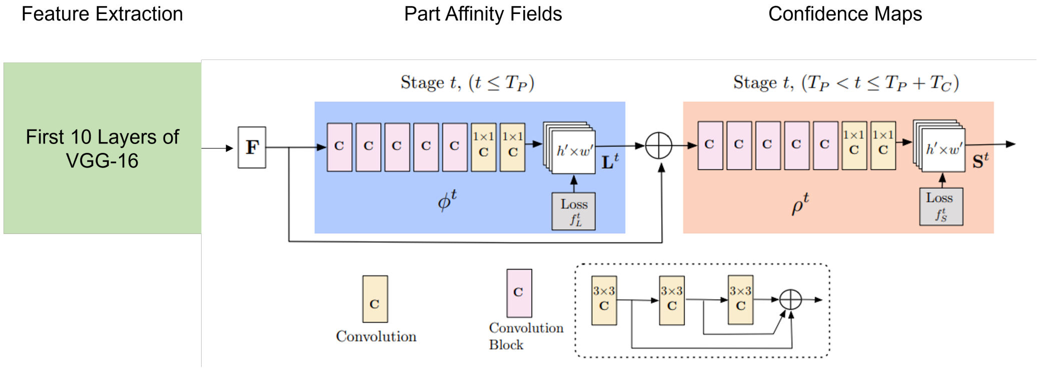


Figure 3. Openpose machine learning model for predicting PAF and confidence maps. The model is composed of three parts; feature extraction, PAF prediction, and confidence map prediction.

The model for predicting these two outputs consists of three convolutional sections. The first section takes input of the image and extracts features. This section is composed of the first 10 layers of the VGG-19 model. The section section takes in these features and outputs the predicted PAFs. Then this output is concatenated with the initial feature inputs and passed through the same model again. This is repeated for T_p times. The total loss uses the outputs for each iteration. The final output is the passed to the final convolutional section which predicts the confidence maps. Similar to the PAF section, the output of this section is concatenated with its input and passed through the model T_C times. The total loss also uses these outputs at each iteration. The final Confidence maps are the final output.

The loss function as mention previously contains all the outputs of each section for every iteration. The loss at each iteration for the PAF and confidence map respectively is:

$$f_L^{t_i} = \sum_{c=1}^{C} \sum_{p} W(p) \cdot ||L_C^{t_i}(p) - L_c^*(p)||_2^2$$

$$f_S^{t_k} = \sum_{j=1}^{J} \sum_{p} W(p) \cdot ||S_j^{t_k}(p) - S_j^*(p)||_2^2$$

Where $W(p)$ is a mask where $W(p) = 0$ if the annotation is not present. $L_C(p)$ & $S_j(p)$ is the predicted PAF and confidence map respectively. $L_c^*(p)$ & $S_j^*(p)$ is the ground truth PAF and confidence map which will be defined later.
The total loss is defined as

$$f = \sum_{t=1}^{T_p} f_L^t + \sum_{t=T_r+1}^{T_p+T_C} f_S^t$$

The ground truths are created from the annotated images present in the dataset. The ground truth confidence maps are constructed by first generating a map for each person and each joint ($S_{j,k}^*$) where j is a specific joint and k is a specific person. Each ($S_{j,k}^*$) is defined as

$$S_{j,k}^*(p) = exp\left(-\frac{||p - x_{j,k}||_2^2}{\sigma^2}\right)$$

Then a final confidence map for each joint is created by taking the max value of all the confidence maps for different people. ($S_j^*(p) = maxS_{j,k}^*(p)$). The ground truth PAF are similarly made by creating a PAF for every person an limb. This is done by considering every pixel in an image. If that pixel is on the person and limb then the PAF vector is the unity vector in the direction of the limb.

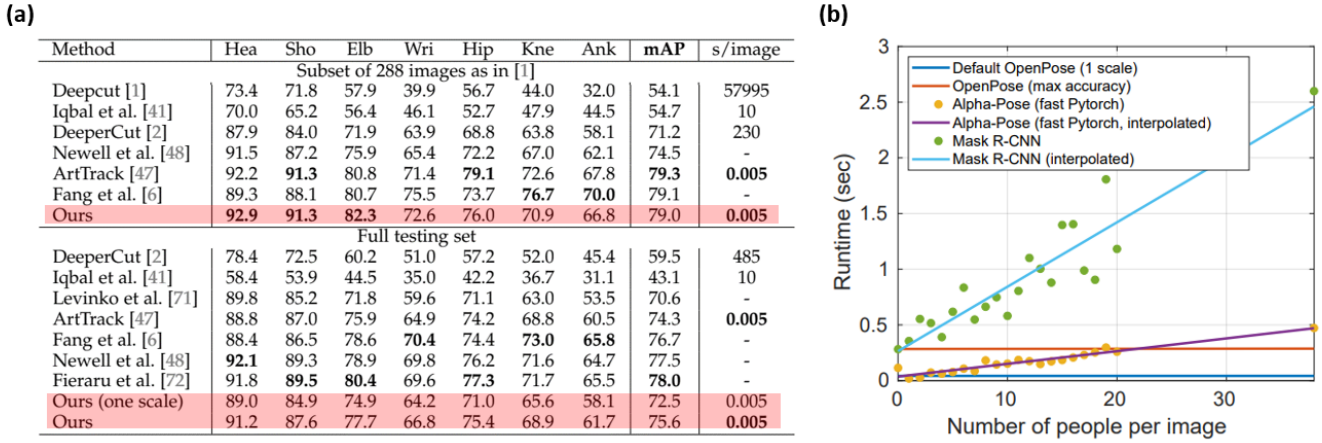
$$L_{c,k}^*(p) = \begin{cases} v & \text{if p on limb c,k} \\ 0 & \text{otherwise} \end{cases}$$

Where c is the limb, k is the person, and v is the unit vector in direction of the limb. The final PAF for each limb is defined as

$$L_c^*(p) = \frac{1}{n_c(p)} \sum_{k} L_{c,k}^*(p)$$

3. Evaluation

The model was evaluated on the MPII dataset using the mean Average Precision (mAP) metric. Figure 4(a) shows a table comparing OpenPose to previous state-of-the-art methods available on the MPII testing sets. It can be seen that OpenPose is capable of performing very comparably to the other methods, even outperforming on some joint detections. The authors also note that their inference time is 6 orders of magnitude less than other techniques. Figure 4(b) shows the results from an inference runtime analysis compared against two popular pose estimation algorithms known as Mask R-CNN and Alpha-Pose. It can be seen that OpenPose does not increase in runtime regardless of how many people are in the image, whereas the other two techniques increase linearly with the number of people in the image. This shows the benefit of the OpenPose approach for use in realtime multi-person pose estimation.



Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
Subset of 288 images as in [1]									
Deepcut [1]	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
Iqbal et al. [41]	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeeperCut [2]	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Newell et al. [48]	91.5	87.2	75.9	65.4	72.2	67.0	62.1	74.5	-
ArtTrack [47]	92.2	**91.3**	80.8	71.4	**79.1**	72.6	67.8	**79.3**	**0.005**
Fang et al. [6]	89.3	88.1	80.7	75.5	73.7	**76.7**	**70.0**	79.1	-
Ours	**92.9**	**91.3**	**82.3**	72.6	76.0	70.9	66.8	79.0	**0.005**
Full testing set									
DeeperCut [2]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al. [41]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Levinko et al. [71]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
ArtTrack [47]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	**0.005**
Fang et al. [6]	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7	-
Newell et al. [48]	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
Fieraru et al. [72]	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**	-
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	**0.005**

Figure 4. (a) Table comparing the performance of OpenPose on the MPII dataset to other approaches; (b) runtime inference analysis comparing OpenPose, AlphaPose, and Mask R-CNN

4. Model Implementation

We ran the pretrained full body model on a few different videos to obtain the annotated human skeletons. One output frame for two of the videos we tested are shown in figure 5. The model did a good job on identifying people in the crowd and approximating their poses. It did a good job on the smaller group as well but there were a few miss identifications. For example, in the center of the room it saw a person that was not there and in the poster on the wall.

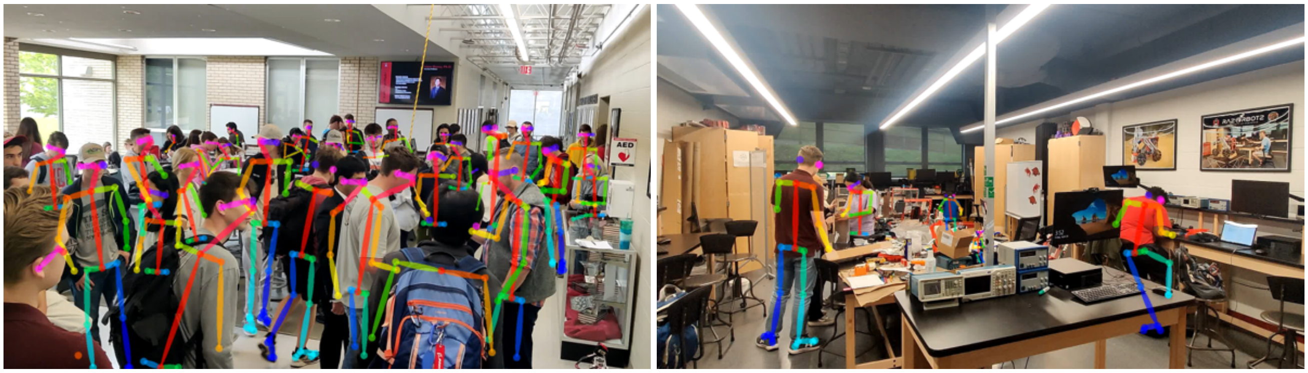


Figure 5. Example outputs of the pretrained model when ran on our own videos.

5. Application

One potential application for human pose estimation is for robotic teleoperation, whereby human motions could be retargeted and used to control a robot. This could be beneficial in situations too adverse for humans to intervene such as search and rescue, or for remote surgeries in the medical field. For this demonstration, we opted to not use OpenPose as we were unable to get above 0.1 fps when trying to run the model in real-time on a webcam. We implement a much more lightweight human pose estimation algorithm developed at Google Research known as Mediapipe [4]. As seen in figure 6, the robot is able to mimic the user's actions through a simple mapping between the estimated human joints to the robot joints. The robot is driven by five 9 g micro servo motors and an Arduino Uno with an external power supply.

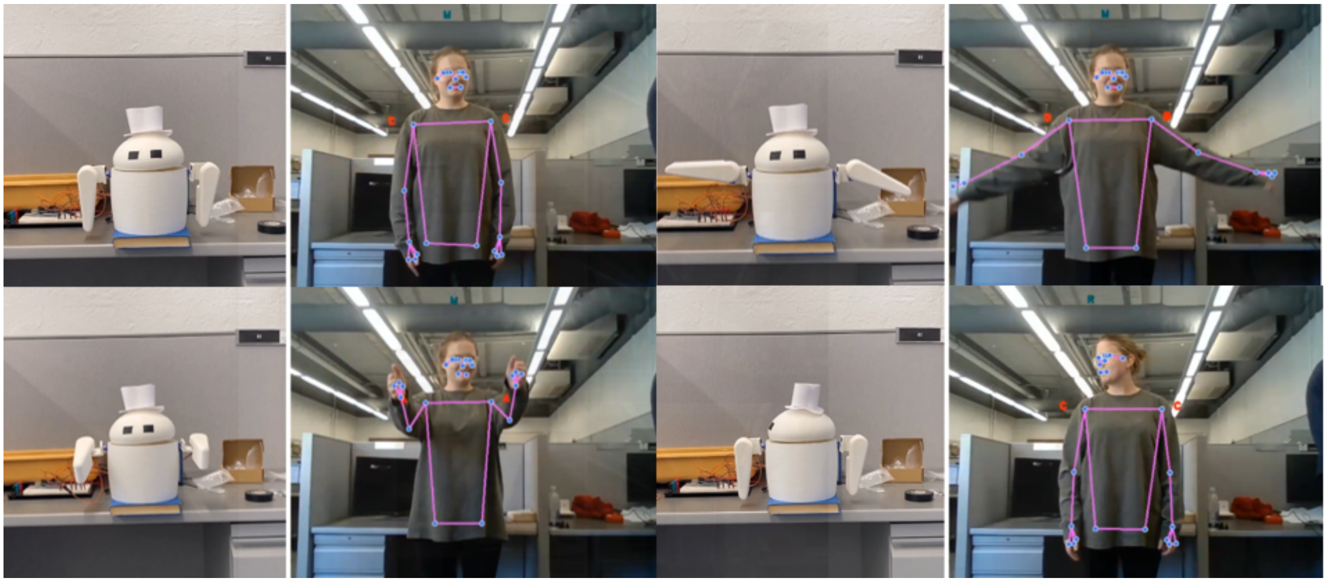


Figure 6. Demonstration of human pose estimation for use in robotic teleoperation

References

[1] Mykhaylo Andriluka et al. "2d human pose estimation: New benchmark and state of the art analysis". In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014, pp. 3686–3693.

[2] Zhe Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: (Dec. 2018). URL: http://arxiv.org/abs/1812.08008.

[3] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[4] Camillo Lugaresi et al. "Mediapipe: A framework for building perception pipelines". In: *arXiv preprint arXiv:1906.08172* (2019).