Investigating Pose Estimation Techniques *

*Submitted in fulfilment for the individual component for the ARI3129 - Advanced Artificial Vision assignment

1st Owen Agius
Department of Artificial Intelligence
Faculty of Information and Communications Technology
University of Malta
owen.agius.19@um.edu.mt

I. Introduction

People detection has long been a popular issue in traditional object detection for a spectrum of uses. Computers can now interpret human body language by performing stance recognition and pose tracking thanks to recent advances in machine-learning techniques. The precision of these detections, as well as the technology needs to execute them, have now reached a commercially viable level.

Furthermore, the technology's development is being deeply influenced by the coronavirus pandemic, where high-performance real-time posture detection and tracking will usher in some of the most impactful advances in computer vision. Combining human position estimation and distance projection algorithms, for example, can be used for social distancing. In a busy environment, it helps people maintain physical space from one another.

When we analyze how pose estimate may be used to automatically detect human movement, we can see how powerful it is. Pose estimation has the potential to develop a new wave of automated systems meant to quantify the precision of human movement, from virtual sports coaches and AI-powered personal trainers to tracking movements on factory floors to ensure worker safety. Autonomous driving is one area where this technology has already proven its worth. Computers can identify and predict pedestrian behavior more accurately with the use of real-time human pose detection and tracking, allowing for more consistent driving.

II. Literature Review

A. Pictorial Structures

A development in an object detection application focusing mostly at faces helped feature detection from photos acquire appeal in the medical industry (Fischler and Elschlager, 1973). Later on, when a new application was built that used a set of descriptors referred to as parts to define face features such as the nose, mouth, and eyes, object detection gained traction for pose estimation (Felzenszwalb and Huttenlocher, 2005). "Spring-like pairwise connections" were used to organize detectable pieces into a deformable structure (Felzenszwalb and Huttenlocher, 2005).

Detecting features on the nose or mouth can be redundant and irrelevant for the implementation of pose estimation, so instead of detecting features on the nose or mouth, body components such as the arms or legs are connected through a series of "spring-like pairwise connections" to denote the body articulations, or joints. The next step was to create a graphical graph that resembled an undirected graph, with the topology specified by the categorical structure of the part locations. The part orientation, position, and foreshortening are used to create the locations.

Convolutional Neural Networks were introduced later, and they outperformed pictorial structures. When evaluating many occluded joints or a posture prediction applied to a complex pose, pictorial structures were found to perform badly. The inefficiency became increasingly frequent as a result of the use of visual structures. Pose estimate algorithms rely largely on the placement of local part descriptors rather than the image's global context.

B. Introduction of Convolutional Neural Networks

DeepPose was the first research to use convolutional neural networks to estimate pose (Toshev and Szegedy, 2014). DeepPose's technique is based on classifying significant points in a posture, which include but are not limited to the major joints. The method is spread out over a convolutional neural network that takes as input the entire image of the pose and computes the features (joints) through each of its hidden layers.

Many people have been inspired by DeepPose's approach, and a variety of 2D pose estimation implementations have been investigated and created as a result (Toshev and Szegedy, 2014). Any of these implementations can be classified as top-down or bottom-up in nature.

Only one pose is fed into the convolutional neural network at a time in the top-down technique. With this in mind, detection must occur early on, detecting the human body and drawing a bounding box around the observed individual.

Because detection occurs on body sections rather than the entire body, the bottom-up technique is less computationally expensive. The discovered physical traits are linked in a graph. The development of the final stance takes place in the last stages. (Insafutdinov et al., 2016), (Li et al., 2019), and (Insafutdinov et al., 2019) are some examples of bottom-up techniques (Cao et al. 2017).

More recently, 3D techniques have been investigated. Although (Nibali et al., 2019) and (Moreno-Noguer, 2017), among others, use this form of architecture, due to a lack of 3D datasets, these tactics are not as widespread as 2D models. In general, 3D posture estimation techniques build on two-dimensional implementations by first capturing and categorizing data in two dimensions, then reconstructing a pose in three dimensions using the results acquired in that dimension space. Single monocular imagery, such as that used by (Nibali et al., 2019) and (Moreno-Noguer, 2017), can be used to simulate three-dimensional structures.

The technique given in allows for three-dimensional rotation and translation from a single 2D image or video (Wattanacheep and Chitsobhuk, 2019). The pose is calculated directly from the 3D model points and the 2D picture points, with inaccuracies being rectified iteratively until an acceptable pose estimate is obtained from a single image (Wattanacheep and Chitsobhuk, 2019). However, multi-view inputs such as marker-based building, which was recently implemented by (Qiu et al., 2019), can be used to represent 3D constructs (Iskakov et al., 2019). Although multi-view models are more accurate than monocular models, they are more expensive to set up due to the additional equipment (cameras and lab) and the time it takes to infer the stance from the markers. A skeleton can be created using the markers. Any recent advances in pose estimation have relied on a skeleton-based model of the human body, such as (Presti and Cascia, 2016) and (Presti and Cascia, 2016). (Liu et al., 2017). This method provides for both 2D and 3D visual input, with 3D data allowing for more depth information and, more importantly for this study, 2D data allowing for a faster and simpler learning process.

The skeleton-based approach is significantly more popular in creating human activity recognition models, as demonstrated by recent work by (Song, et al., 2017), which offers human activity recognition using a deep learning strategy of building an RNN to generate a spatio-temporal 3D skeleton model. (Ji et al., 2013) and (Zhao et al., 2017) advocate for the adoption of a 3D CNN rather than the RNN recommended by (Song, et al. ,2017).

III. Methodology

A. *BlazePose*

The method employs a two-step detector-tracker machine learning process, which first locates the person/pose ROI (Region of Interest) within the frame. Using the ROI-cropped frame as input, the tracker then predicts the posture landmarks and segmentation mask within the ROI. The detector is only used as needed in the video use cases, that is, for the first frame and when the tracker can no longer detect the presence of a body posture in the preceding frame.

For other frames, the pipeline simply calculates the ROI based on the pose landmarks from the previous frame. The pipeline is implemented as a MediaPipe graph that renders using a specific posture renderer sub-graph and uses a pose landmark sub-graph from the pose landmark module. A pose detection sub-graph from the pose detection module is used internally by the pose landmark. The detector is inspired from the BlazeFace model (Bazarevsky et. al., 2019), used in MediaPipe Face Detection, as a proxy for a person detector.

The COCO topology, which comprises of 17 markers spanning the torso, arms, legs, and face, is the current standard for human body posture. The COCO keypoints, only localize to the ankle and wrist points, leaving hands and feet without scale and orientation information, which is critical for practical applications like fitness and dancing. More keypoints are required for the use of domain-specific pose estimation models, such as those for hands, faces, and feet. BlazePose introduces a novel topology with 33 human body keypoints that is a superset of the COCO, BlazeFace, and BlazePalm topologies. This enables the identification of body semantics based just on pose prediction, which is consistent with face and hand models.

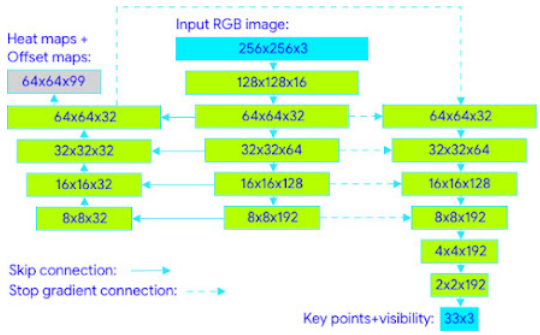

Fig. 1: Tracking network architecture (Bazarevsky et. al., 2019)

Each component of the whole learning pipeline, which includes pose detection and tracking models, must be extremely fast, taking only a few milliseconds per frame, to achieve real-time performance. To do so, it is observed that the person's face sends the strongest signal to the neural network concerning the position of the body.

All 33 person keypoints with three degrees of freedom (x, y location, and visibility), as well as the two virtual alignment

keypoints indicated above, are predicted by the posture estimation component of the pipeline. Rather than using a compute-intensive heatmap prediction, the model uses a regression technique supervised by a combined heat map/offset prediction of all keypoints, as illustrated in Fig. 1. Specifically, a heatmap and offset loss are used to train the network's center and left towers during training. The heatmap output is then removed, and the regression encoder (right tower) is trained, effectively employing the heatmap to supervise a lightweight embedding.

B. MoveNet

MoveNet is a bottom-up pose estimate algorithm that uses heatmaps to locate human keypoints accurately. There are two parts to the architecture: a feature extractor and a set of prediction heads. The estimation technique is based on CenterNet, but with a few tweaks that increase speed and accuracy. The TensorFlow Object Detection API is used to train all of the models.

MobileNetV2 with an attached feature pyramid network (FPN) is MoveNet's feature extractor (as shown in Fig. 2) , allowing for a high-resolution (output stride 4) and semantically rich feature map output.

COCO and an internal Google dataset named Active were used to train MoveNet. While COCO is the industry standard for detection due to its wide range of scenes and scales, it is not ideal for fitness and dance applications, which feature hard poses and severe motion blur. Active was created by identifying keypoints on YouTube yoga, exercise, and dance videos (using COCO's standard 17 body keypoints). To foster diversity of scenes and persons, only three frames from each video are chosen for training.

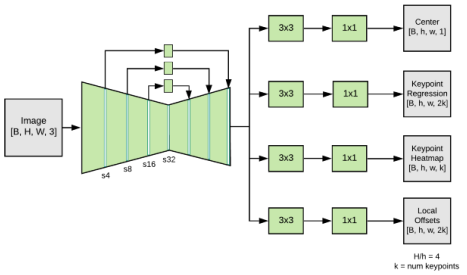

Fig. 2: Source: https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html

As shown in Fig. 3, the MoveNet framework outputs a "person center heatmap, a keypoint regression field, a person keypoint heatmap alongside a 2D per-keypoint offset field" [18].

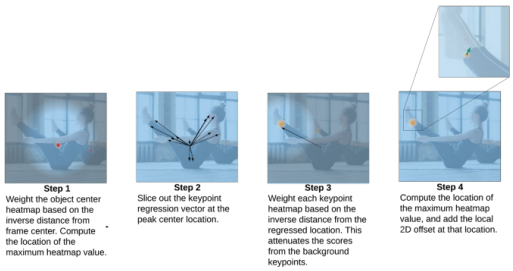

Fig. 3: Source: https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html)

MoveNet consists of two sub-implementations being MoveNet-Thunder and MoveNet-Lightning. In this experiment MoveNet-Thunder was chosen ahead of MoveNet-Lightning. Although the prior is deemed to return better results, a slighly 'poorer' performing implementation was chosen to be evaluated against the well-established framework 'BlazePose'. The Thunder model is most suitable for detecting the pose of a single person who is 3ft to 6ft away from the camera.

IV. Evaluation

Both implementations were evaluated using two different videos, where each video portrays a much different action than the other. The first video contains a very wide range of movement as a person is attempting to kick a ball. Whereas, the second video is simply of a person walking in a straight line.

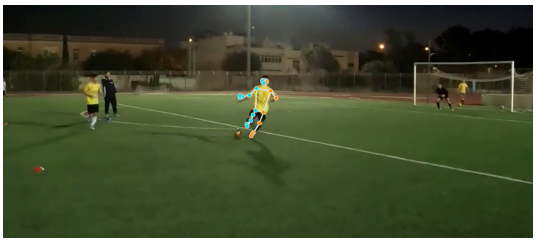

Fig. 4: BlazePose Capture (Source: Owen Agius)

As can be seen in the images above (Fig. 4 and Fig. 5a), the BlazePose implementation performs better coming to accuracy of pose estimation at a high rate of movement with the detected key-points remaining clear throughout the whole movement. The MoveNet framework gets confused throughout the video as the detected person is found to be a person in the background (As viewed in Fig.2b) rather than the person who the detection started out on.

In order to extenuate the error of estimation provided by the MoveNet framework, the implementation was run on a single image with a sole figure to be pose estimated. As can be seen from Fig. 6b, the MoveNet pose estimation performs

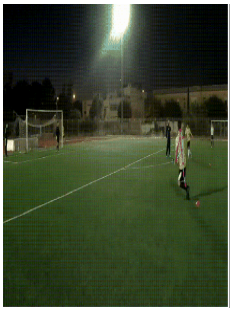


Fig. 5: Progression of MoveNet on a Video (Source: Owen Agius)

quite poorly. The framework detects the ball as another key-point of the figure alongside skipping entirely the figure's right knee and creating a link between the elbow and the ankle. BlazePose performs extremely well for the same image (Fig. 6a) where all joints and links are highlighted perfectly to the image.

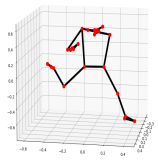


Fig. 6: MoveNet and BlazePose on a Static Image (Source: Owen Agius)

Considering, a video of a simpler movement, both implementations perform greatly. This nature of video plays more into the hands of the MoveNet-Thunder implementation which is deemed to perform better when the subjects are single and 3-6 feet away from the camera. Although, both poses are detected accordingly (Fig 7. and Fig 8.), BlazePose still keeps an edge over MoveNet. This is solely due to the fact that the BlazePose implementation implements 33 different key-points as opposed to the standard 17 COCO body key-points presented by MoveNet.

The final test between these two implementations is to check the performance whenever the figure is behind an object which occludes certain key-points. The image provided consists of a person walking but in this scenario, the lower part of the body is being covered up by a push-chair. In contrast to the trend in the earlier tests, MoveNet performs much better. Considering, Fig 9., BlazePose fails to construct the occluded areas of the body, but classifies correctly the areas that are visible. This

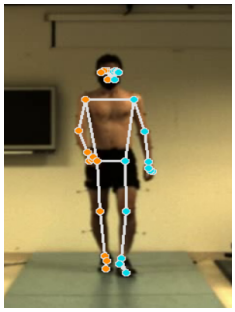

Fig. 7: BlazePose Capture (Source: Owen Agius)

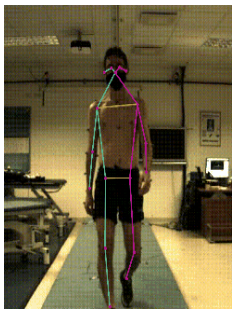

Fig. 8: MoveNet Capture (Source: Owen Agius)

is not the case for the MoveNet framework (Fig. 10), which does not only construct properly the visible key-points, but also predicts with a high accuracy the occluded area of the body.

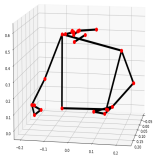

Fig. 9: BlazePose Capture (Source: Owen Agius)

Fig. 10: MoveNet Capture (Source: Owen Agius)

V. Conclusion

After thorough experimentation on the BlazePose and MoveNet frameworks, it can be concluded that although the BlazePose implementation worked better for most experiments, the MoveNet framework performs better when certain parts of the body are occluded.

The BlazePose architecture is made up of the detection of 33 key-points rather than the 17 classic COCO key-points in use by the MoveNet framework. This enhancement in the architecture reflects in the overall performance of the BlazePose implementation. MoveNet was outperformed by the latter in the scenarios of regular and more rigorous movements.

Further experimentation and comparison between these frameworks can be performed by implementing more mathematical error metrics such as, PCP, PCK, PDJ, MPJPE and AUC. All in all, the results eccrued by both of the frameworks were rendered to be satisfactory and as expected.

References

[1] Piemontese, F. (2019). "Markerless Motion Analysis from Synchronized 2D Camera Views: A Convolutional Neural Network Approach". Thesis, University of Padova.

[2] Fischler, M. A. and R. A. Elschlager (1973). "The Representation and Matching of Pictorial Structures". In: IEEE Transactions on Computers 1, pp. 67–92.

[3] Felzenszwalb, P. F. and D. P. Huttenlocher (2005). "Pictorial Structures for Object Recognition". In: International Journal of Computer Vision 61.1, pp. 55–79.

[4] Toshev, A. and C. Szegedy (2014). "DeepPose: Human Pose Estimation via Deep Neural Networks". In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653, 1660.

[5] Insafutdinov, E., L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele (2016). "DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model". In: European Conference on Computer Vision. Springer, pp. 34–50.

[6] Li, J., C. Wang, H. Zhu, Y. Mao, H. Fang, and C. Lu (2019). "CrowdPose: Efficient Crowded Scenes Pose Estimation and a New Benchmark". In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10855–10864.

[7] Cao, Z., T. Simon, S. Wei, and Y. Sheikh (2017). "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299.

[8] Nibali, A., Z. He, S. Morgan, and L. Prendergast (2019). "3D Human Pose estimation with 2D Marginal Heatmaps". In: IEEE Conference on Applications of Computer Vision, pp. 1477–1485.

[9] Moreno-Noguer, F. (2017). "3D Human Pose Estimation from a Single Image via Distance Matrix Regression". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2823–2832.

[10] Wattanacheep, Bhattarabhorn and Chitsobhuk, Orachat. (2019). "Prediction of 3D rotation and translation from 2D images. ICCCM 2019: Proceedings of the 2019 7th International Conference on Computer and Communications Management". 49-52.

[11] Qiu, H., C. Wang, J. Wang, N. Wang, and W. Zeng (2019). "Cross View Fusion for 3D Human Pose Estimation". In: IEEE International Conference on Computer Vision, pp. 4342–4351.

[12] Iskakov, K., E. Burkov, V. Lempitsky, and Y. Malkov (2019). "Learnable Triangulation of Human Pose". In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7718–7727

[13] Presti, L. L., and La Cascia, M. (2016). 3D skeleton-based human action classification: A survey. Pattern Recognition, 53, 130-147.

[14] Chen, Y., Shen, C., Wei, X. S., Liu, L., and Yang, J. (2017). Adversarial posenet: A structure-aware convolutional network for human pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1212-1221).

[15] Ince, O. F., Ince, I. F., Park, J. S., and Song, J. K. (2017, June). Gait analysis and identification based on joint information using RGB-depth camera. In 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 561-563). IEEE.

[16] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., and Grundmann, M. (2020). BlazePose: On-device Real-time Body Pose tracking. arXiv preprint arXiv:2006.10204.

[17] Bazarevsky, V., Kartynnik, Y., Vakunov, A., Raveendran, K., and Grundmann, M. (2019). Blazeface: Sub-millisecond neural face detection on mobile gpus. arXiv preprint arXiv:1907.05047.

[18] https://medium.com/axinc-ai/movenet-pose-estimation-for-video-with-intense-motion-2b92f53f3c8