Analyzing embedded pose estimation solutions for human behaviour understanding

José Gomes da Silva Neto
Universidade Federal de Pernambuco
Voxar Labs
Recife, Brazil
jgsn@cin.ufpe.br

João Marcelo Xavier Natário Teixeira
Universidade Federal de Pernambuco
Voxar Labs
Recife, Brazil
jmxnt@cin.ufpe.br

Veronica Teichrieb
Universidade Federal de Pernambuco
Voxar Labs
Recife, Brazil
vt@cin.ufpe.br

***Abstract*—This work represents the first phase of a more complete work that has the goal of using RGB images as information to make analyses of human behavior. In this phase, we developed a prototype of hardware/software, capable of estimating human pose using only RGB information. The equipment chosen was the NVIDIA Jetson Nano, known for having have a better computational performance compared to Raspberry pi and Arduino microcontoller alternatives. In the search for important algorithms for pose estimation, applied to the limited platform as the Jetson Nano, we found important works such as HyperPose, TensorRT Pose Estimation, and the used on the project, tf-pose-estimation. The results show a low FPS performance of the Jetson Nano, using the chosen algorithm, compared to related hardware, such as the NVIDIA Jetson TX2 and NVIDIA Jetson Xavier.**

***Index Terms*—Jetson Nano, RGB, Human pose Estimation, NVIDIA**

I. Introduction

Computer vision techniques that make use of depth images are a growing area of research, mainly because of the vast possibilities of applications, including security, industry, marketing, and mobile robots and cars. But to use these techniques, it is essential to have more high-level hardware, such as RGB-D cameras.

Search for an alternative that facilitates the use of this kind of information, since depth capture devices are make expensive than rgb cameras, our project aims to develop and evaluate a prototype of hardware plus software capable of capturing RGB images and generating a report of the behavior of the people tracked in the images. This solution can be essential in security applications because most security cameras are only RGB, so they do not have depth information.

This paper will present the first part of the project. The tracking of people using RGB images. For that phase, the initial point was to choose the hardware. That has to be small enough to make the solution portable and practical. It requires a high computational performance because of the analytics of the scene. It is essential to have deep learning as a possible solution. To do that, the hardware should have a good CPU and GPU performance. The conclusion analyzing all the needs was the open-source NVIDIA Jetson Nano. It has a 4GB 64-bit LPDDR4 @ 25.6GB/s of memory RAM and a Quad-Core Arm A57 @ 1.43 GHz processor. That shows it is a more suitable choice the computer Raspberry Pi or an the Arduino microcontroller .

The remainder of this paper is structured as follows. Section 2 list some of the related works found. Section 3 details the hardware chosen. Section 4 describes the methodology adopted while section 5 talks about the results obtained. At last, Section 6 concludes this first phase of the project.

II. Related Work

This project has two important parts: Hardware and Software. After choosing the hardware, we focused on making a search on state-of-the-art algorithms for human pose detection. In this search it is important that the algorithms have the capacity to be used on a limited execution platform, as the Jetson Nano. An important search made by our work is looking for works where is compared the performance of the Jetson Nano with other hardwares, using different algorithms, that will give us one idea of what is the performance level expected by the Jetson Nano.

To estimate human pose (in 2D or 3D) refers to locate the anatomical key parts of the individual ([5], [2], [1], [7]). Estimating the pose of multiple people using images, mainly where they are doing some specific activity, is really challenging. Each image may have an unknown number of people in any position and size. A normal approach to this problem ([8], [6]) is to use a person detection algorithm and after a pose estimation algorithm for each detected person. These Top-Down approaches are related with works that focus on pose detection for one person([3], [10]).

The first analysed work was [9], in which the authors propose a neural network and a tensor decomposition for pose estimation. In this work it is possible to do the detection of poses in real time for multiple people. The test was made using an NVIDIA Titan Xp GPU, and it showed a 30PFS frame rate.

Another important work is [4], which was the first real-time multi-person system to jointly detect human body, hand, face, and foot keypoints (in total 135 keypoints) on single images. The test achieved about 22 FPS running in a machine with an NVIDIA GTX 1080 Ti.

To refine the search, we started looking for algorithms tested on Jetson Nano or similar hardware. One of the initial results

Identify applicable funding agency here. If none, delete this.

was tf-pose-estimation [1], a human pose estimation algorithm implemented using Tensorflow. It also provides several variants that have some changes to the network structure for real-time processing on the CPU or low-power embedded devices, that make it a very important work in our context. This algorithm was tested on Jetson Nano TX2 and Macbook Pro 15".

Another important work foung is TensorRT Pose Estimation [2]. This project features multi-instance pose estimation accelerated by NVIDIA TensorRT. It is ideal for applications where low latency is necessary. This work was tested in NVIDIA Jetson Nano and NVIDIA Jetson Xavier, which makes it very important to have an expectation about the performance of NVIDIA Jetson Nano compared with other hardware.

A most recent project is HyperPose [3], a library for building human pose estimation systems that can efficiently operate in the wild. This work is important because it makeperformance comparasion with the tf-pose estimation cited before.

III. Hardware detailing

It is crucial to evaluate the performance of the Jetson Nano, compared to other possible hardware. Following the NVIDIA Jetson line, we have two hardware, which are already used as a benchmark for the Jetson Nano, in the works cited in the related works section. They are NVIDIA Jetson Xavier and NVIDIA Jetson TX2, and their specifications are shown in table I. The corresponding performances are shown in Table II.

IV. Methodology

In the first phase of this project, the goal is to develop a hardware and software solution, capable of estimating human pose. For that was chosen the NVIDIA Jetson Nano as the hardware and the algorithm chosen was the tf-pose-estimation, to be a variation of Open Pose, a widely used algorithm in the human pose estimation task.

The tf-pose-estimation has a few important dependencies: Python3; Tensorflow 1.4.1+; opencv3; protobuf; python3-tk; slidingwindow. After installing all these dependencies, you can install the tf-pose-estimation library. These dependencies highlight the need to have hardware of high considerable computational performance.

It is important to highlight the three possible model graphs: mobilenetThin (trained in 432x368); mobilenetV2Large (trained in 432x368); mobilenetV2Small (trained in 432x368). Each model graph can be used, but some can demand more computational performance than another, and by consequence, the least demanding are less accurate. The input for tf-pose-estimation are of two types: Images or video. The outputs are a heatmap, Vectormap-x; Vectormap-y; Pose estimation, as seen in Figure 1, for images, and an FPS information with the pose estimation for video.

[1]https://github.com/ildoonet/tf-pose-estimation
[2]https://github.com/NVIDIA-AI-IOT/trt_pose
[3]https://github.com/tensorlayer/hyperpose

The tests were performed in two phases: Images as input; Video as input, for all these phases was evaluated the performance, in this case, the FPS, and the accuracy. For the case they were using images as input we used images with multi-person and people with occlusion of some part of their body. It also was tested on four different graph models, to check how was the real impact of using each one. The performance results were also compared with the benchmark provided by the project.

V. Results

The results are divided into two phases: Images as input and Video as input. In the first case, we will show only the result of the pose estimation, because it is the most important information for our project.

A. Image as Input

In this phase, we chose three types of pictures: 1. One person without occlusion; 2. One person with occlusion; 3. Multiple people without occlusion; 4. Multiple people with occlusion. The results obtained are shown in Figures 2, 3, 4 and 5.

B. Video as Input

In this phase, we evaluate based on FPS, to each graph model, to determine how is the performance of each one in the Jetson Nano, and be able to compare that with the performance of the Jetson TX2. The performance results are shown in Table III.

C. Trained models Evaluation

In this test, we evaluated the accuracy of each one of the three possible Trained modelss available on tf-pose-estimation algorithm: 1. MobilenetThin; 2. mobilenetV2Large; 3. mobilenetV2Small. To evaluate the accuracy, we ran each Trained models with the same images. The chosen images are with one people without occlusion and multiple persons without occlusion. The results can be seen in Figures 6, 7, 8, 9, 10 and 11.

VI. Conclusion

This work is the first phase of a project that has the goal of developing a smart camera applied to human behavior recognition. This initial phase focused on improving the prototype of hardware and software, capable of estimating human pose. The hardware chosen was the NVIDIA Jetson Nano for its good computational performance and being portable and practical. After that, we did a review on the state-of-art of works of human pose estimation to find an algorithm capable of being used on a limited platform of development. The test focuses on evaluating the algorithm performance on Jetson Nano.

It is possible to conclude in the case where images are used as input, the algorithm deals well with situations without occlusion, even in multi-person cases. But in situations with occlusion, the performance is very compromised. Where videos are used as input, the FPS is very low compared with other hardware; this was already expected due to the work

TABLE I
COMPARATIVE HARDWARE'S SPECIFICATION

Specification	Hardware		
	Jetson TX2	*Jetson Xavier*	*Jetson Nano*
AI Performance	**1.33 TFLOPs**	**6 TFLOPs**	**427 GFLOPs**
GPU	**256-core NVIDIA Pascal**	**512-Core GPU with Tensor Cores**	**128-core NVIDIA Maxwell**
CPU	**Dual-core Denver 2 64-bit**	**8-Core ARM v8.2 64-Bit**	**Quad-Core ARM Cortex-A57 MPCore**
Memory	**4 GB 128-bit LPDDR 51,2GB/s**	**32 GB 256-Bit LPDDR4x — 137 GB/s**	**4 GB 64-bit LPDDR4 25,6GB/s**

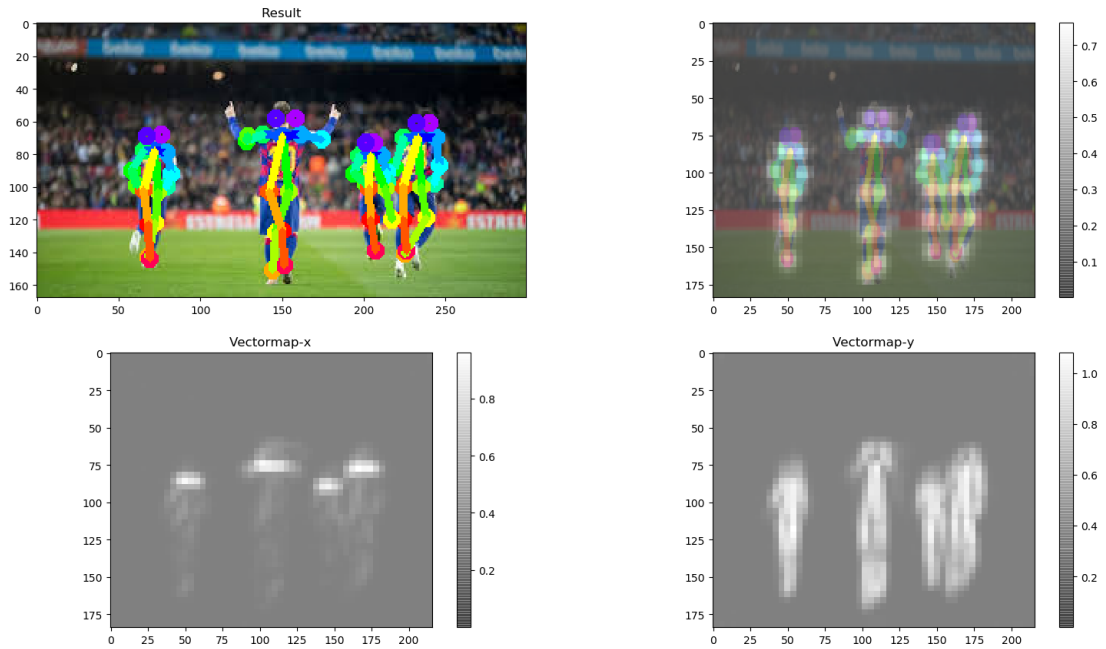

Fig. 1. Tf-pose-estimation output.

TABLE II
COMPARATIVE HARDWARE'S PERFORMANCE

Algorithm	Hardware		
	Jetson TX2	*Jetson Xavier*	*Jetson Nano*
tf-pose-estimation	**10 FPS**	**-**	**-**
TrT Pose Estimation	**-**	**251 FPS**	**22 FPS**

TABLE III
JETSON NANO PERFORMANCE ON TF-POSE-ESTIOMATION ALGORITHM.

Hardware	*Trained models*	*Performance*
Jetson Nano	**MobilenetThin**	**1.6 - 2.9 FPS**
Jetson Nano	**mobilenetV2Large**	**1,8 - 2.5 FPS**
Jetson Nano	**mobilenetV2Smal**	**1.5 - 3.9 FPS**

findings in the state-of-art review. It is possible to improve the FPS frame rate, changing the model graphic, but the accuracy will be compromised. The FPS performance also can be improved using other possibles algorithms as the HyperPose. The witch has a performance ten times better compared with

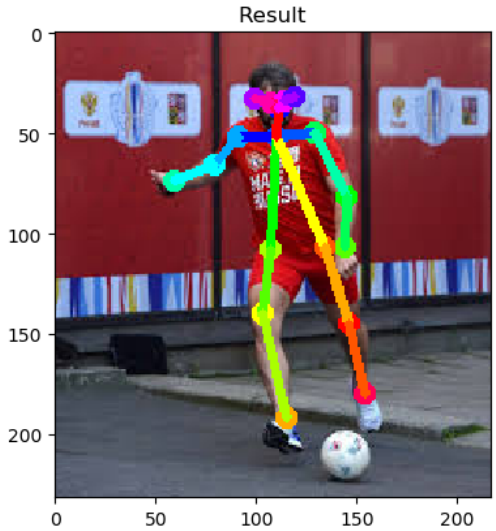

Fig. 2. One person without occlusion.

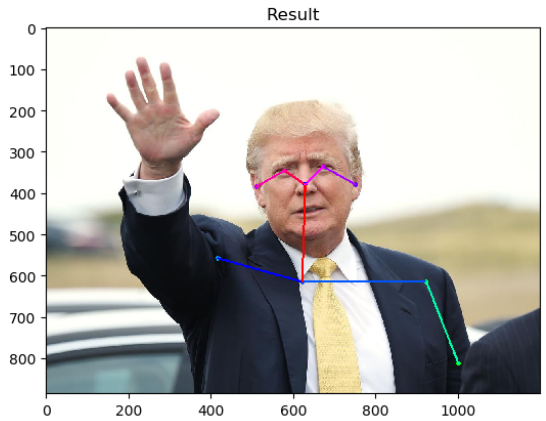

Fig. 3. One person with occlusion.

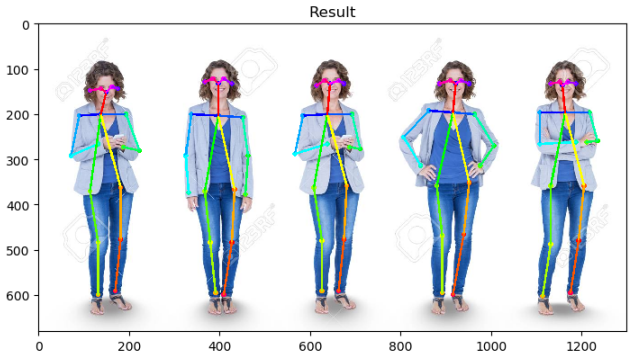

Fig. 4. Multiple persons without occlusion.

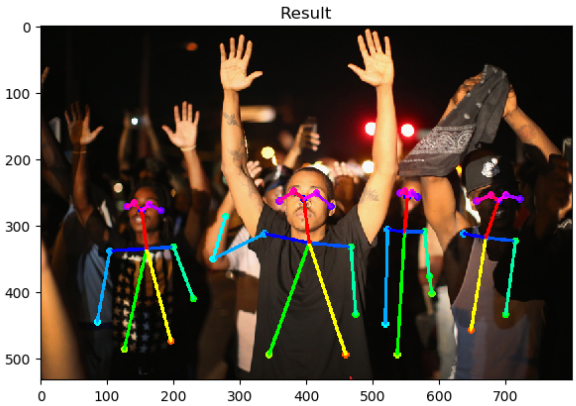

Fig. 5. Multiple persons with occlusion.

the tf-pose-estimation.

The next steps of the project are to improve the algorithm performance, if not possible, test other available algorithms, and begin the smart phase of the project. That means associate the human pose estimation in real-time with artificial intelligence to make possible behavior analytics.

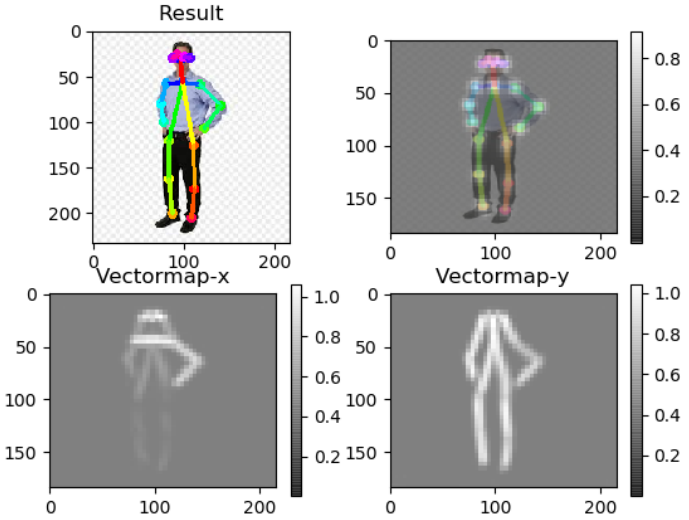

Fig. 6. tf-pose-estimation running with MobilenetThin Trained models .

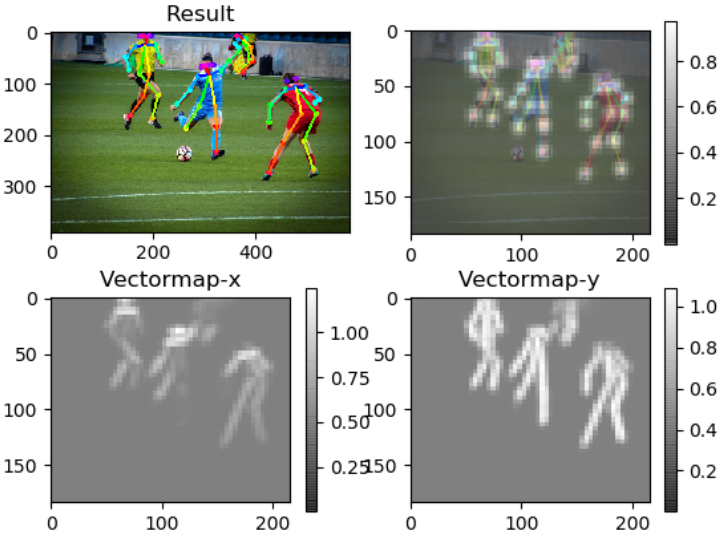

Fig. 7. tf-pose-estimation running with MobilenetThin Trained models.

References

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1014–1021. IEEE, 2009.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010.

[3] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 468–475. IEEE, 2017.

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

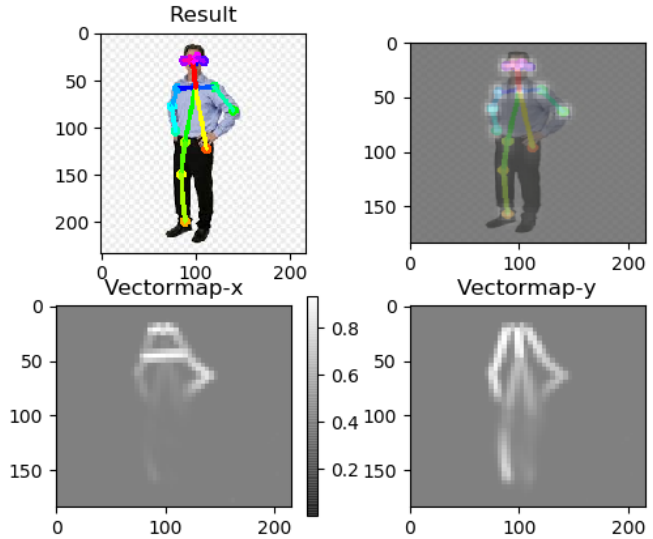

Fig. 8. tf-pose-estimation running with mobilenetV2Large Trained models.

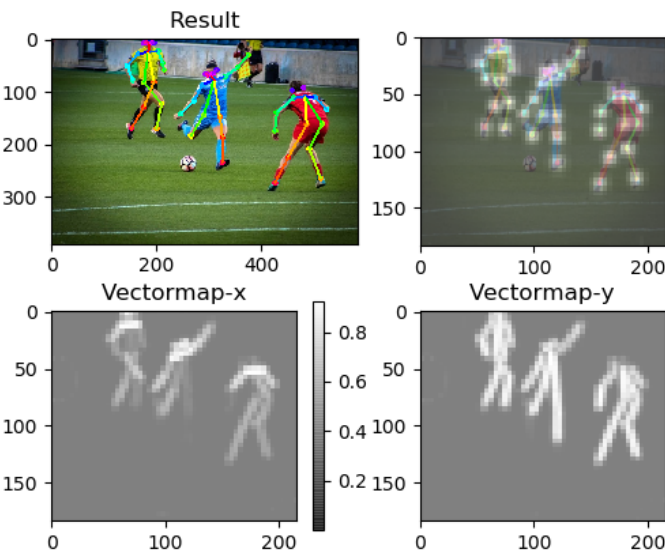

Fig. 9. tf-pose-estimation running with mobilenetV2Large Trained models.

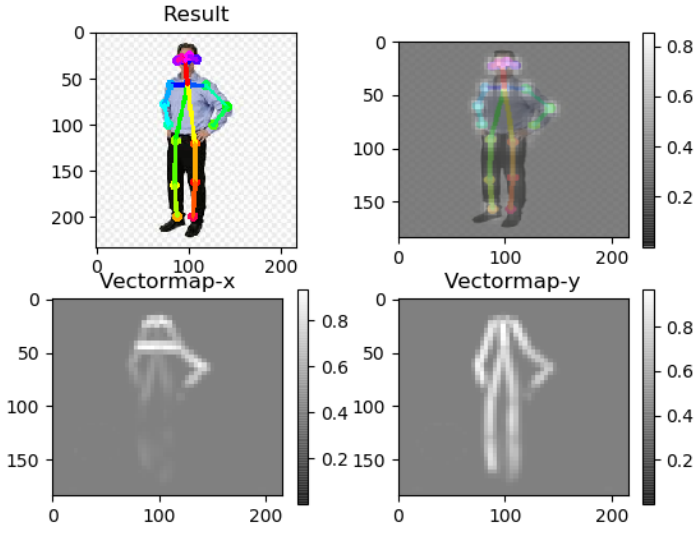

Fig. 10. tf-pose-estimation running with mobilenetV2Small Trained models.

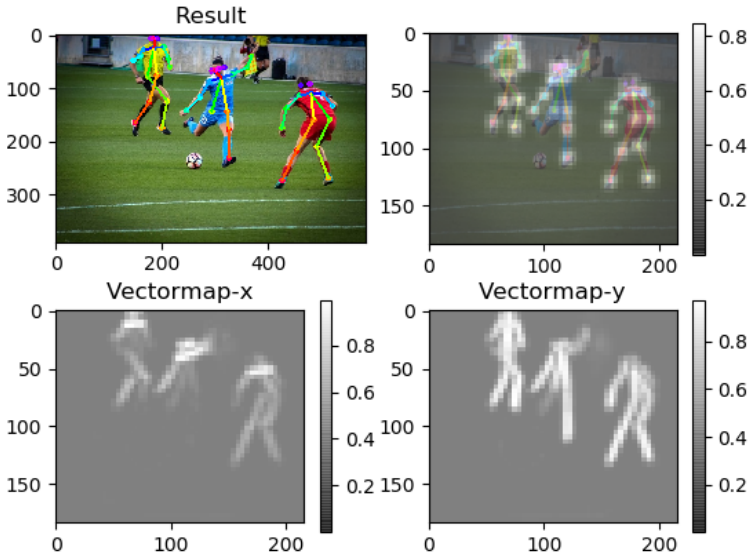

Fig. 11. tf-pose-estimation running with mobilenetV2Small Trained models.

[5] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[6] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. Using k-poselets for detecting people and localizing their keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3582–3589, 2014.

[7] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[8] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185. IEEE, 2012.

[9] Luiz José Schirmer Silva, Djalma Lúcio Soares da Silva, Alberto Barbosa Raposo, Luiz Velho, and Hélio Côrtes Vieira Lopes. Tensorpose: Real-time pose estimation for interactive applications. *Computers & Graphics*, 85:1–14, 2019.

[10] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.