CIVIL-456 Milestone 3 Report

Group 10 : Cavedon Nicolas, Duranceau Agatha, Gieruc Théo, Ranglaret Baptiste

ABSTRACT

This report presents the final Milestone of the Loomo project for the course CIVIL-456 Deep Learning for Autonomous Vehicles. Our goal is to implement a person identification and tracking on the Loomo robot, for a human-robot tandem race. We use several neural networks for human detection, pose classification, tracking, re-identification and up-scaling.

1 INTRODUCTION

The goal of this project is to understand how state-of-the-art deep learning methods can be used for perception in autonomous vehicles, in this case a Loomo robot. Here, human-robot tandems must participate in a race. To do so, the Loomo must recognise its human as a person of interest (POI), track and follow them until they reach the final goal.

2 ARCHITECTURE

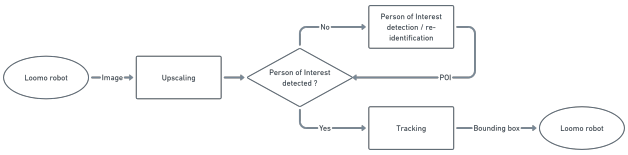

Figure 1: The global architecture of the project

Figure 1 shows a representation of the global architecture of the project. Before entering the main part of the program, the images sent from the Loomo robot are first up-scaled using Real-ESRGAN [4]. For more information, please refer to section 5. The program is composed of a Finite State Machine (FSM) which has the following states:

The default state of the FSM is the *searching* state. The program continuously searches for a person of interest (POI) using OpenPifPaf [2] to detect a sequence of poses. For more information, please refer to section 3. Once a person of interest is successfully detected, the FSM switches to the *tracking* state. To get the correct ID for the person to track, the program uses the Jaccard index (also called Intersection over Union) between the OpenPifPaf bounding box of the detected POI with the one from YOLOX. Then, the tracking uses a combination of YOLOX [1] and DeepSORT [5] [6] to track the POI. During tracking, a sequence detection is done every second to be able to switch who the program is tracking. When the first pose of the selected sequence is detected, the program goes into the searching state until either the sequence is finished, or is reset by timeout. The program also switches back to search state if the POI is lost.

3 PERSON OF INTEREST DETECTION

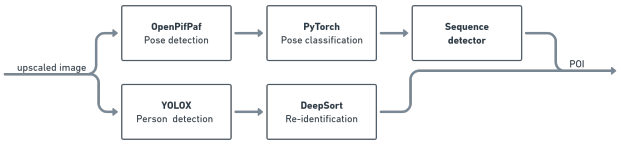

Figure 2: The (re)detection of the person of interest

The detection of the POI is the central point of the architecture as can be seen on figure 1. The details of the procedure of (re)identification are shown in figure 3. First, the upscaled image is analyzed and the pose of each person in the frame is detected using OpenPifPaf algorithm. The joints used for classification are the elbows and the shoulders as the sequence only requires arm movements. The pose of each person is then classified using a simple PyTorch neural network, trained with a custom dataset. Only "known" poses are kept (i.e. poses in the training set). These poses are then sent to the sequence detector, that keeps track of the sequence of poses that each person in the frame has achieved. If the sequence chosen as "password" is achieved by one of the persons in the frame, then this person becomes POI.

4 TRACKING THE PERSON OF INTEREST

For tracking the POI, we have chosen to use YOLOX (yolox-s model before the race) because of its high classification accuracy coupled with a relatively high speed, which made it interesting for real-time applications [1]. In practice, it ran much faster than with OpenPifPaf, and as pose detection is required only for the sequence detection step, once the ID of the POI was known, YOLOX was sufficient to return the bounding box of the POI to be tracked by DeepSORT.

DeepSORT is an extension of SORT and is one of the most commonly used tracking networks. It uses a Kalman filter and assumes that the agents have linear velocities to predict their position in the next frame. It then computes a distance metric to see which newly detected agent corresponds to the predicted bounding box. However, this method alone is not robust to occlusion and is not capable of performing re-identification. This is why DeepSORT also computes an appearance feature vector for each bounding box, to be able to continue tracking the objects after they have been temporarily lost, or if two objects temporarily share approximately the same position (which could lead to a switch of the IDs of the two objects if it did not consider the object features) [3]. The result can be observed in figure 3.

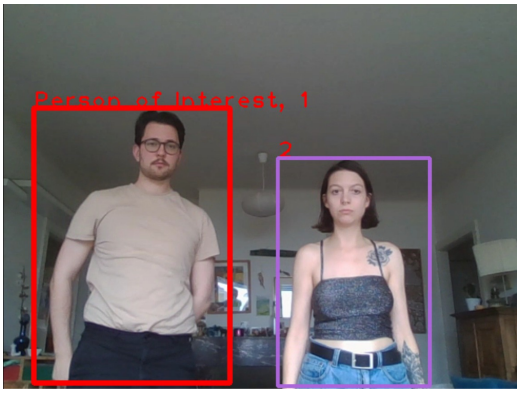

Figure 3: Identification and tracking of the POI against other persons

5 UPSCALING

The Loomo robot sends images with a resolution of 160x120 pixels, which is too small for good detection (YOLOX has been trained on 640x640 pixels images [1]). Therefore, in order not to have to re-train the model, we have decided to use an up-scaler. To select the best one, we have compared several models: first using OpenCV (see Figure 4), then using deep nets (see Figure 5).

We opted for realESRGAN [4] (see figure 5), which has better results than openCV (figure 4 while being the fastest among Neural Networks up-samplers. To further enhance the image, we have also sharpened it with a convolution of kernel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (1)$$

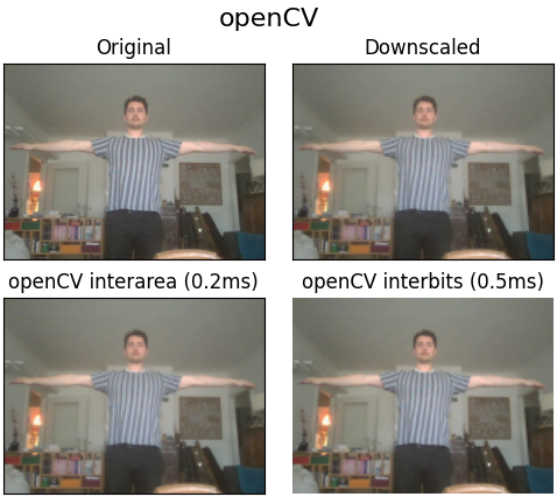

Figure 4: Upscaling on OpenCV

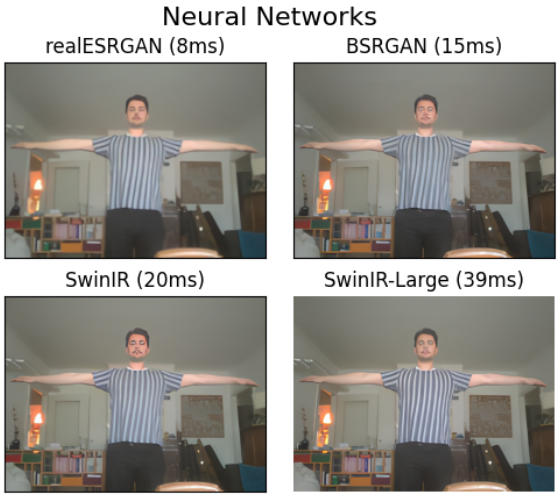

Figure 5: Upscaling using deep nets

6 THE RACE

During the race, our algorithm under-performed compared to the exercise sessions. As it affected the POI detection as well as the tracking, we could pinpoint the source of the problem : the person detection, done with YOLOX. In order to have good inference time, we had privileged the use of the smaller pre-trained model *YOLOX-s* (9.0M parameters) . However, in the race lighting conditions, that model wasn't performing well enough for having reliable detection and tracking. This was due to the glass roof which made silhouettes harder to recognise against the light compared with the weekly tests performed in a regular corridor. Switching to *YOLOX-m* (25.3M parameters) solved that problem and allowed us to finally participate in the race.

7 CONCLUSION

The architecture of our project allowed us to successfully detect, identify and track a person of interest by using state-of-the-art deep learning methods. As we have realised during the race, scene characteristics may play a huge role in the performance of an autonomous vehicle, and using more complex, robust models is key when working in an unpredictable, changing environment.

REFERENCES

[1] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430* (2021).
[2] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. 2021. OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Transactions on Intelligent Transportation Systems* (March 2021), 1–14.
[3] Shishira R Maiya. 2019. DeepSORT: Deep Learning to Track Custom Objects in a Video. *Nanonets* (2019). https://nanonets.com/blog/object-tracking-deepsort/
[4] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. [n.d.]. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In *International Conference on Computer Vision Workshops (ICCVW)* (2021).
[5] Nicolai Wojke and Alex Bewley. 2018. Deep Cosine Metric Learning for Person Re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 748–756. https://doi.org/10.1109/WACV.2018.00087
[6] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple Online and Realtime Tracking with a Deep Association Metric. In *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962