Data and Artificial Intelligence for Transportation Pattern Tracking

Yannick Paul Klose
Yi Zhiyuan
Department of Computer Science, EPFL , Switzerland

***Abstract*—The aim of this project is to design a tracking algorithm that is able to identify a predefined pattern in an unknown environment. The algorithm will be used to automatically control a *Loomo* robot to follow a person with the predefined pattern. For this specific task a neural network architecture based on the opensource library *openpifpaf* is used. In order to improve the model data exploration, image expansion and hyper parameter search is applied.**

I. Introduction

Over the last few years, the computing power of GPU's has increased rapidly enabling us to compute complex operations in parallel at high speed. As a result with the use of Big Data, Machine Learning and especially the use of convolutional neural networks is becoming increasingly important.

Idenitfing objects in complex images plays a fundamental role in Machine Learning and is already used in many ways to optimise processes. The goal of this project is to modify an existing state of the art Machine Learning algorithm, named *openpifpaf*, that is used for multi-person 2D human pose estimation to identify predefined patterns on images.

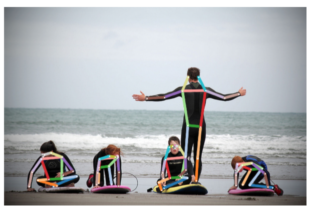
This report provides an overview of the steps needed to modify the given architecture and the challenges faced during the process. Section I describes the idea of modifying the input of the model, Section II describes methods to improve the model with data augmentation and Section III finally describes the end result and the tests on the *Loomo* robot.

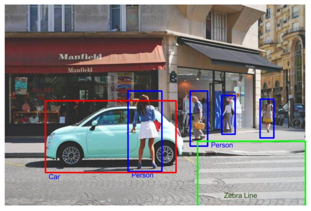
II. Idea

At a first glance an observation of the existing architecture is needed. The state of the art architecture *openpifpaf* is based on the keypoint detection. That means the algorithms take keypoints such as ”left-shoulder” or ”right leg” as an input and tries to learn these specific points for multiple objects an image. The information about the keypoints and many other parameters are given through the OpenSource Coco Dataset.

The first step is to modify the algorithm to a general object detection task that sets bounding boxes around the detected objects instead of a human skeleton. The expected output is shown in figure 1.

In order to achieve this idea, the data information from the Coco Database, stored in a .json file had to be modified while keeping in mind that the modified input still has to be Coco Keypoints. The idea is to access the bounding box information stored in the database, then determine the center of the box to finally use this point as input for the underlying architecture. As a result each Coco-Keypoint will present one bounding box, which can be interpreted as one object.


(a) openpifpaf [1]



(b) object detection [2]

Fig. 1: multi pose estimation with openpifpaf and expected object detection

The problem faced with this method is the fact, that the training takes a long time and in the end the model will just identify different objects in an image. However, the desired goal is to detect only one particular object with great certainty. Therefore the next approach will only focus on one specific class.

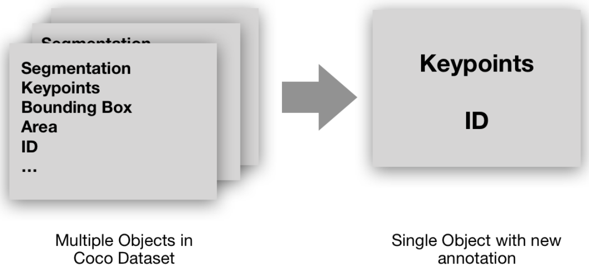

Segmentation
Keypoints
Bounding Box
Area
ID
...
Keypoints
ID
Multiple Objects in Coco Dataset
Single Object with new annotation

Fig. 2: new annotation, example for one image

In this approach a specific and unique pattern is used to paste on background. The idea is that the center of the pasted image will be converted into a Coco Keypoint which is then used in the same matter as the previous approach as input for the model. In order to achieve this, the MS-Coco Database had to be modified while loading the images into the DataLoader. Usually the structure of the MS-Coco Database consists of multiple elements such as ”Category”, ”Keypoints”, ”ImageID”, ”Segmentation”, ”Bounding Box” and more. Furthermore each image can contain multiple of these elements, because most of the images consist of multiple objects. However for the desired task only one element is needed, consisting of only a ”Keypoint” and ”ID”. Therefore a new annotation is created that takes exclusively ”Keypoint” and ”ID” into account (figure 2)[3]. This means that the input is completely uncorrelated to the given data in the original Coco Database, because the Coco images are only used as background noise.

III. Increasing performance techniques

After training a model, based on the idea described in Section I, the result was already quite good. However, it turned out that the model only performs well under certain, very similar conditions. In order to generalise the model, data augmentation and image expansion is needed.

According to research, it is important not only to insert the object to be examined at random positions, but also to change its appearance. Therefore the following parameters were considered as important: scaling, rotation, Gaussian blur, brightness and random positioning. The determination of the parameters is based on several factors. First of all, the object should be in a similar size ratio as it will be from the point of view of a robot. The use of a Gaussian blur was used to create as realistic conditions as possible to simulate fast movements with insufficient frame rate.

Fig. 3: Loomo Robot [4]

With the improved model described above, a first test attempt with the *Loomo* robot has been made. It turned out that the model already performed well. A more detailed analysis showed that the camera of the robot displays bright colours darker than they were. For this reason a new model was trained, which now had a modified, darker pattern to track. In addition the size of the pattern that should be recognised by the model was decreased since the camera lens of the *Loomo* robot appeared to be a wide lens.

In addition the dataset was expended by own images, that were taken at the place where the robot should do its final test. Because the number of images are relativly small, again data augmentation was used to generate more images. Since this technique was considered just as fine tuning and should definitely not result in overfitting, the previously trained model was taken as input for the fine tuned one. In total the training on the new dataset took only 5 percent of the total training time.

In a final step the detector of the robot was modified in such a way that if the model does not predict anything, the model will take the prediction of the last successful prediction. To not result in a high errors this prediction recovery was capped at a maximum of 5 frames.

IV. Final results

After training the model on another different pattern, it was tested again on *Loomo*. The results of detection improved visually a lot according to the training with darker colours. The result can been seen in figure 4. The data is send to the server (V100 Machine) in order to compute the prediction. Afterwards the robot receives the predicted keypoint which should represent the center of the desired tracking object.

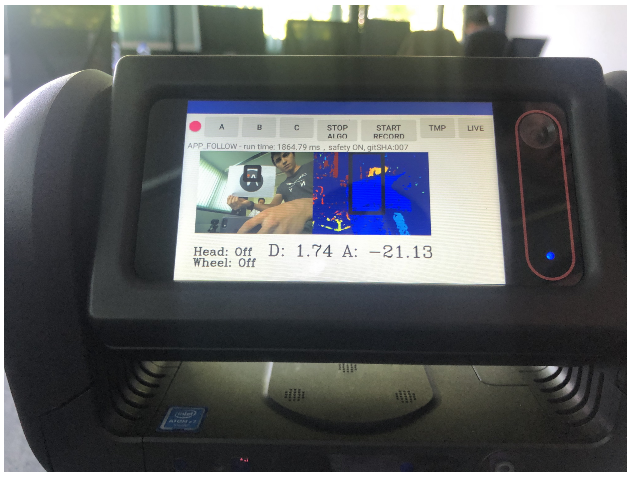


Fig. 4: Tests on pattern detection

V. Conclusion

As a conclusion the project successfully showed the use of convolutional neural networks as a powerful tool for object detection. Data augmentation and hyperparameter search have proven to be important methods of improvement. In addition, a powerful server is required for a smooth transmission to calculate the predictions in real-time. After applying these methods it was possible to create an object detection algorithm for a *Loomo* robot (figure 3).

References

[1] S. Kreiss, L. Bertoni, and A. Alahi, “Pifpaf: Composite fields for human pose estimation,” *CVPR, arXiv preprint arXiv:1903.06593*, 2019.

[2] A. Sachan, “Picture object detection,” 2019, https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/.

[3] A. Kelly, “Create coco annotations from scratch,” 2019, http://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch.

[4] L. Robotics, “Picture loomo robot,” 2019, https://gaminggadgets.de/loomo-segway-wird-zum-robotik-spielzeugjp-carousel-35858.

[5] S. Kreiss and A. Alahi, “Epfl course - data and artificial intelligence for transportation,” 2019, cIVIL-459.