POVa – Traffic sign detection and recognition

Kateřina Fořtová (xforto00)
Richard Letanec xletan00)
Ondřej Pospíšil (xpospi0a)

ABSTRACT

Traffic sign detection is an important task in the autonomous driving system. We chose the pretrained Faster R–CNN detection model, which we trained on the Mapillary dataset. Dataset contains annotated images with more than 300 classes, we processed the dataset and categorized all traffic signs into 5 basic classes – warning, information, regulatory, complementary and other-sign. We used the trained model to predict traffic signs on videos.

Keywords: POVa, FIT, VUT, traffic signs, Faster R-CNN, Mapillary

INTRODUCTION

The goal of this project is detection and classification of traffic signs in video using existing detector. In the project we will train models from Tensorflow 2 Detection Model Zoo repository and predict detection model not only on videos but also on images. Repository is available on GitHub also with links to Colab file training and Drive folder of trained models checkpoints and results (see file `README.md` in project GitHub repository)[1].

IMPLEMENTATION

Dataset

The Mapillary dataset containing annotated traffic signs images from all over the world was chosen to train the model[2]. The training dataset used for our task consists of 12197 images, the validation dataset then contains 5320 images. Traffic signs are classified into more than 300 classes (e.g. *warning–roundabout–g25*, *information–bus-stop–g1*, *regulatory–maximum-speed-limit-65–g2*, *complementary–keep-right–g1* or *other-sign*). For our needs, classification into such a large number of categories would be disadvantageous and therefore the dataset is preprocessed and objects are classified into only 5 basic classes – *warning*, *information*, *regulatory*, *complementary* and *other-sign*.

Faster R-CNN

Faster R-CNN is an object detection approach that was introduced in 2016. It improves and solves the problems of its predecessors – R-CNN (2014) and Fast R-CNN (2015). First of all, the problem of the selective search algorithm typical for these two previous approaches used for recursive merging of similar regions into larger regions is solved. This algorithm affects the performance of the neural network, is not capable of the learning process and was therefore canceled in Faster R-CNN model. In Faster R-CNN, the entire image is fed to the convolutional neural network input and a feature map is generated. However, the model also includes a second separate network for predicting a suitable areas for detection – region proposals. These predicted suitable areas are then reshaped using the ROI Pooling layer, the class of the bounding box and its offset values are predicted [1].

[1] `https://github.com/kateriska/POVa-project`
[2] `https://www.mapillary.com/dataset/trafficsign`

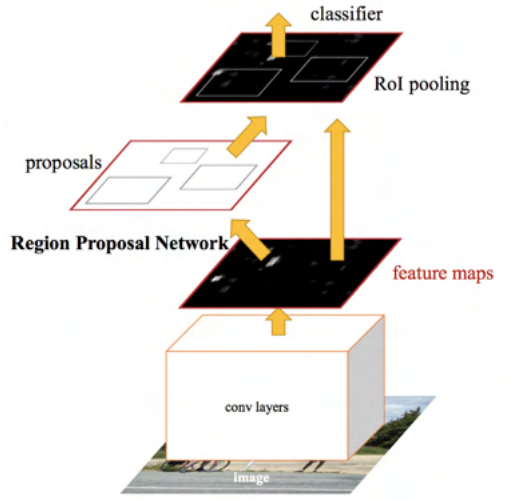


Figure 1. Principle of Faster R-CNN [1]

Detection Model

Faster R-CNN ResNet50 V1 640x640 was used as detection model for this project. This model is part of the TensorFlow 2 Detection Model Zoo repository of TensorFlow Model Garden available under the Apache 2.0 license[3]. All models from TensorFlow Model Garden are pretrained on COCO 2017 dataset[4] which is one of the datasets widely used for machine learning tasks. Originally, the Single Shot Detector – SSD model (SSD MobileNet V2 FPNLite 640x640) was also considered, but the Faster R-CNN was preferred for the final training due to the generally higher accuracy even with the cost of higher training time. The Colab Pro environment using T4 and P100 GPUs was used for training.

Each model from repository is composed of pre-trained checkpoint and `pipeline.config` file which is used for setting parameters of detection model such as count of training steps, batch size or last checkpoint configuration. Mapillary training and validation datasets are transformed into TF records. TF records contain byte representation of each image, positions of bounding boxes, annotated classes of bounding boxes, image width, image height and other information.

Graphs of the development of different types of losses for the final Faster R-CNN model depending on the parameter `num_train_steps`:

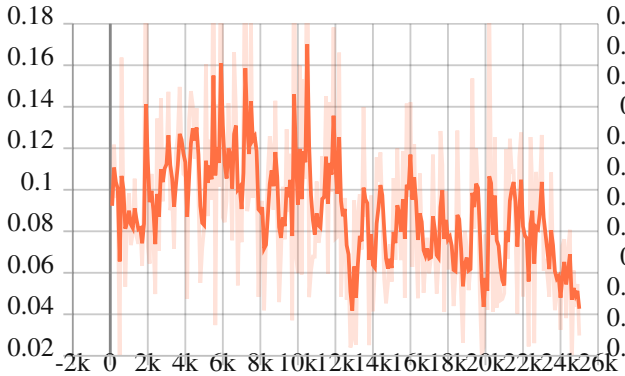


Figure 2. BoxClassifierLoss – classification loss

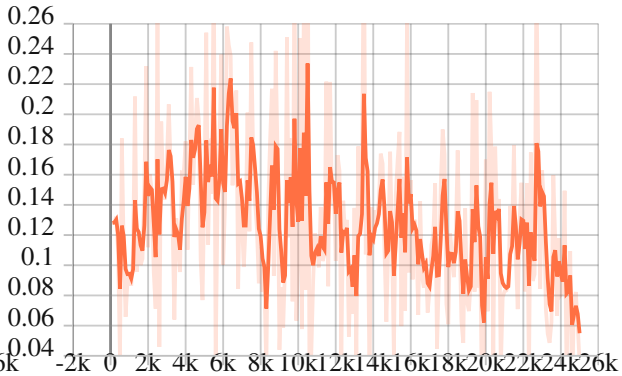


Figure 3. BoxClassifierLoss – localization loss

[3] https://github.com/tensorflow/models
[4] https://cocodataset.org

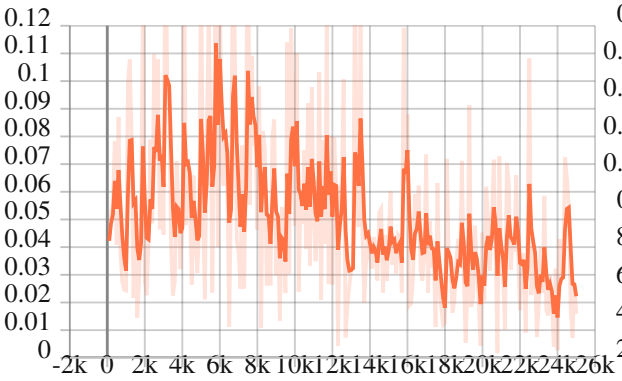


Figure 4. RPNLoss – localization loss

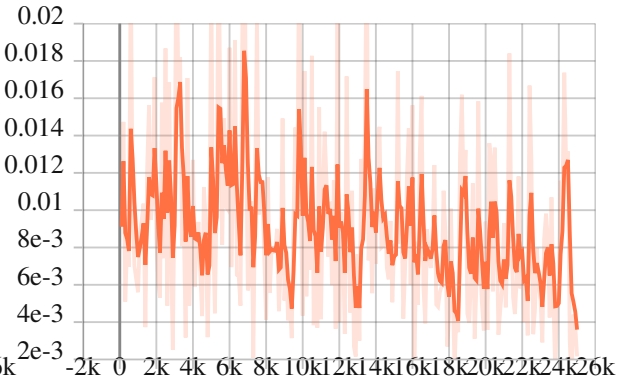


Figure 5. RPNLoss – objectness loss

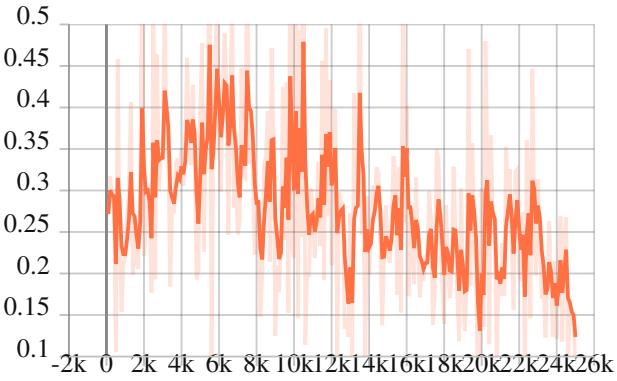


Figure 6. Loss – total loss

Image Detection and Prediction with Trained Model

For each entry, the model predicts the 100 bounding boxes with the highest certainty score. Only 20 bounding boxes with a minimum detection score of 40 % are selected. In many cases, the certainty score for classification into a given class is between 60 – 90 % for trained Faster R-CNN model. The model is able to classify into 5 basic categories. Images from several sources were used for testing: the German test dataset (The German Traffic Sign Detection Benchmark [5]) – due to the high similarity of the traffic signs with the Czech Republic, images from the test Mapillary dataset and images taken by the authors of the project.

Video Detection

We process video frame by frame. For each frame of the video model predicts the bounding boxes, together with classes and scores. This basic approach proves to be noisy, we propose the prediction which count with the detection from the previous frame.

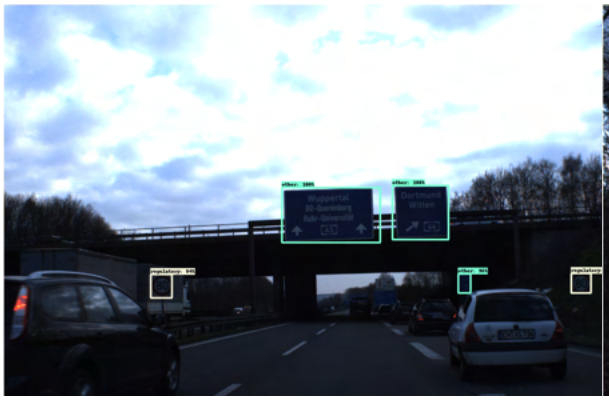
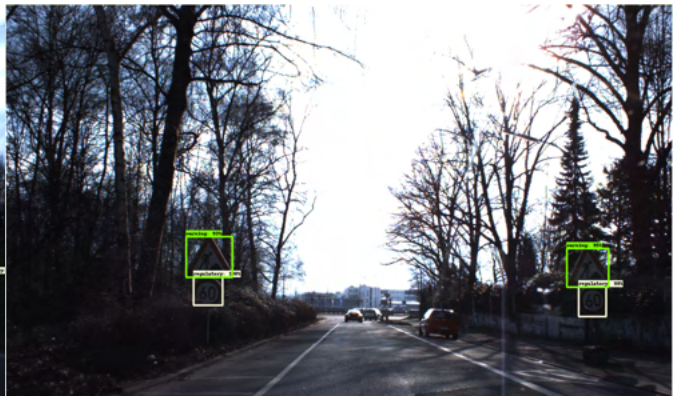
First we determined if bounding boxes in one frame overlap with boxes in second frame. We used the *Intersection over Union* metric. The metric provides the overlap ratio of the two boxes. We chose the IoU threshold rather small, if there is an overlap larger than 10 % its the same traffic sign. For each pair of the matched boxes we decide which class to use, for that we use the maximum of the scores. We choose the class with the bigger score, but the bounding box is still the same.

CONCLUSION

In this project we managed to train Faster R-CNN model for detection and classification of traffic signs from images and video. We trained the model on images from Mapillary dataset and were able to achieve certainty score for classification between 60 – 90 %. We also tried to train SSD model, which achieved

[5] `https://benchmark.ini.rub.de/gtsdb_news.html`

lower score. For the task of traffic sign detection in video we implemented an application, that can detect and classify traffic signs in video using our trained models. We proposed an approach to improve the class prediction by using the detection from two latest frames, if the bounding boxes in those frames overlap, and choosing the class with higher score. Further work in this project could be done by implementing a separate model for sign classification, which might potentially increase the accuracy of the predicted classes.





REFERENCES

[1] Gandhi, R. (2018). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.