Appendix

In this appendix we list the external libraries we used, and show a few extra results for which there was not enough space left in the actual work. Please note, however, that since this is breaking the page limit, we do not expect any content of the appendix to count towards our grades.

A. External code

We implemented all parts of the pipeline except the 3D point triangulation ourselves. For 3D point triangulation we used COLMAP[2]. To write data into the COLMAP database we used the *COLMAP Database script*[3], which is part of the COLMAP project code. Other than that we used OpenCV[4]—mainly for template matching—and other python libraries like NumPy[5], Matplotlib[6], SciPy[7] scikit-learn[8], and Transforms3d[9].

B. Traffic Sign Detection based on Deep Learning

[2] https://github.com/colmap/colmap
[3] https://github.com/colmap/colmap/blob/dev/scripts/python/database.py
[4] https://opencv.org/
[5] https://numpy.org/
[6] https://matplotlib.org/
[7] https://www.scipy.org/
[8] https://scikit-learn.org/
[9] https://github.com/matthew-brett/transforms3d

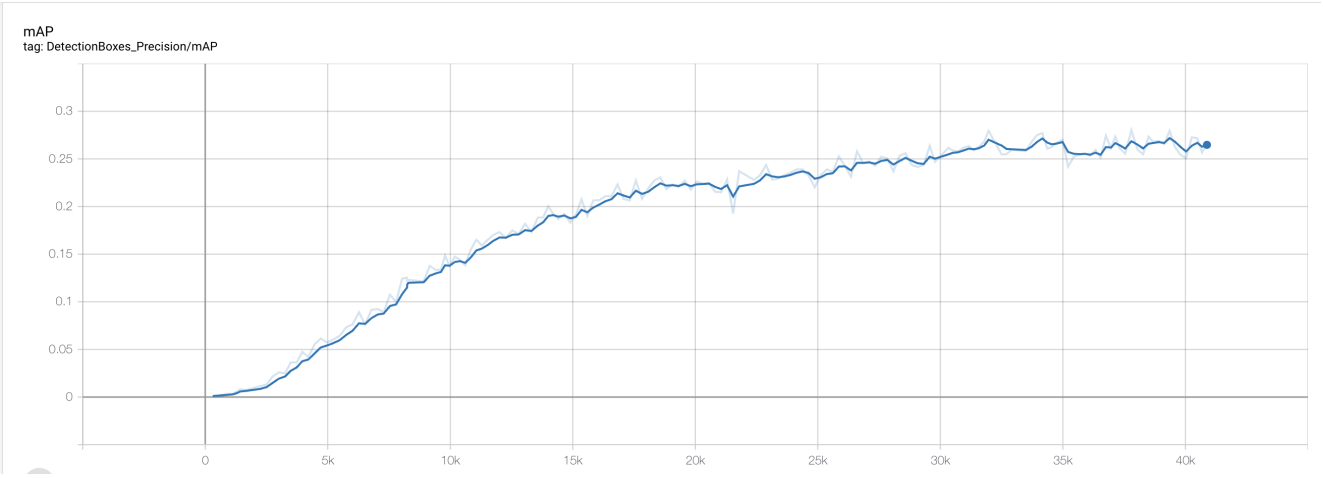

Figure 8: Mean average precision after 40,000 epochs, detecting 42 classes on the GTSDB dataset using a faster_rcnn_inception_v2 archeticture.

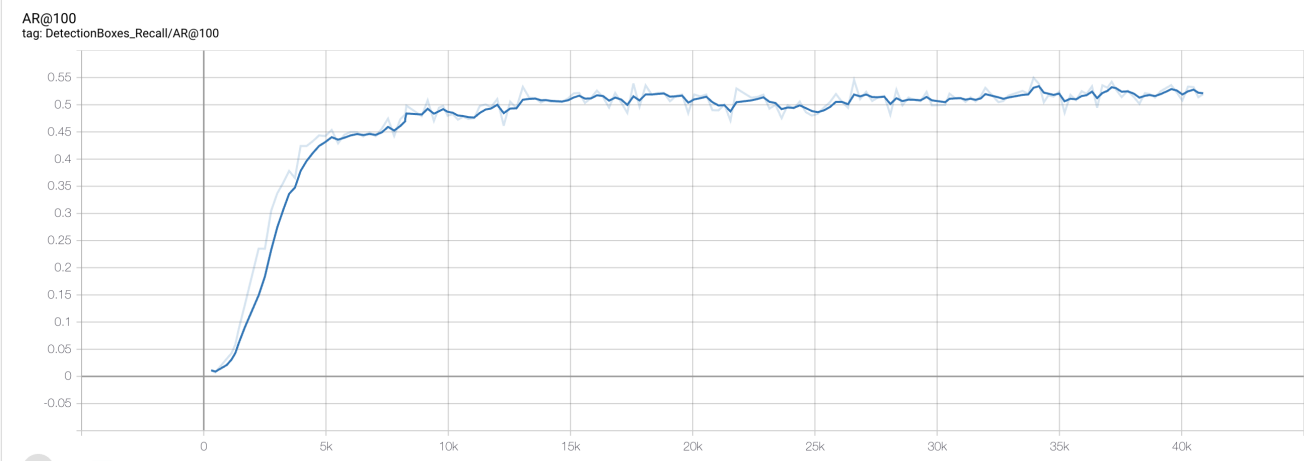

Figure 9: Mean average precision after 40,000 epochs, detecting 42 classes on the GTSDB dataset using a faster_rcnn_inception_v2 archeticture.

C. Failure cases of template matching

Figure 10: Sequence of frames where false positive detections appear. In this case the shape of the tree branches creates a triangular shape that our object detector matches to the yield sign template.