FAKULTA STROJNÍHO INŽENÝRSTVÍ ústav automatizace a informatiky

COMPUTER VISION FOR AUTONOMOUS VEHICLES

Autor: Michal Lečbych
Vedoucí: Mhd Ali Shehadeh

DIPLOMOVÁ PRÁCE, 2022

Objectives

1. Familiarise with a technological basis of computer vision used in autonomous systems in self-driving cars.
2. Create your own neural network model.
3. Build a custom computer vision system.

Introduction

Automation systems, which would require minimal or no human intervention for self-driving cars, have seen a significant increase in research. New methods based on machine learning have been created, providing very good results. This work focuses primarily on object detection-based perception systems that do not require additional complex hardware and could potentially be used in simple devices, contributing to additional safety for many drivers.

Scene understanding

The goal is to understand the surroundings of a car and get a compact representation of it. The more information we can get from the world, the better the autonomous system can be built. These systems are usually complex systems made of numerous tasks. The particular interest is taken for object detection-based systems. To do this, a custom neural network model was created. The distance estimation model for car instances and lane detection system was developed.

Object detection model

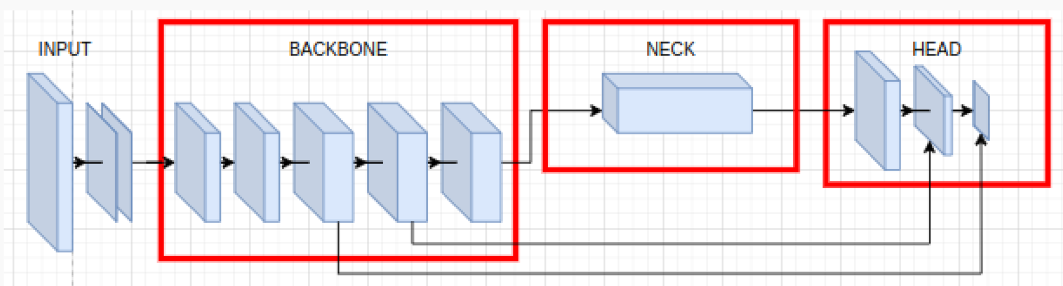


Figure: The model architecture

- This new model includes methods such as Mish activation, larger kernel size (7x7), or grouped convolutions in the backbone part.
- The neck part is a pooling layer, called spatial pyramid pooling. The purpose of the neck block is to add additional layers between the backbone and the head.
- The head part's goal is to output the coordinates of the bounding box (x, y, w, and h) and the confidence score for each class. This is done on 3 different scales, so the best fitting bounding box could be chosen.
- This neural network was trained from scratch on PASCAL VOC dataset. Consisting of 16,000 images for the training dataset and 4000 images for the test dataset.

Distance estimation

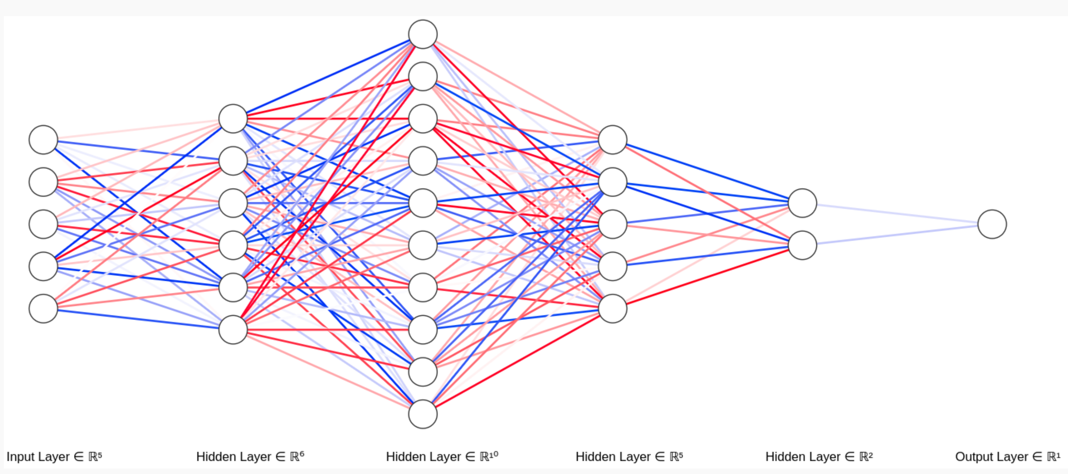


Figure: Feed forward network

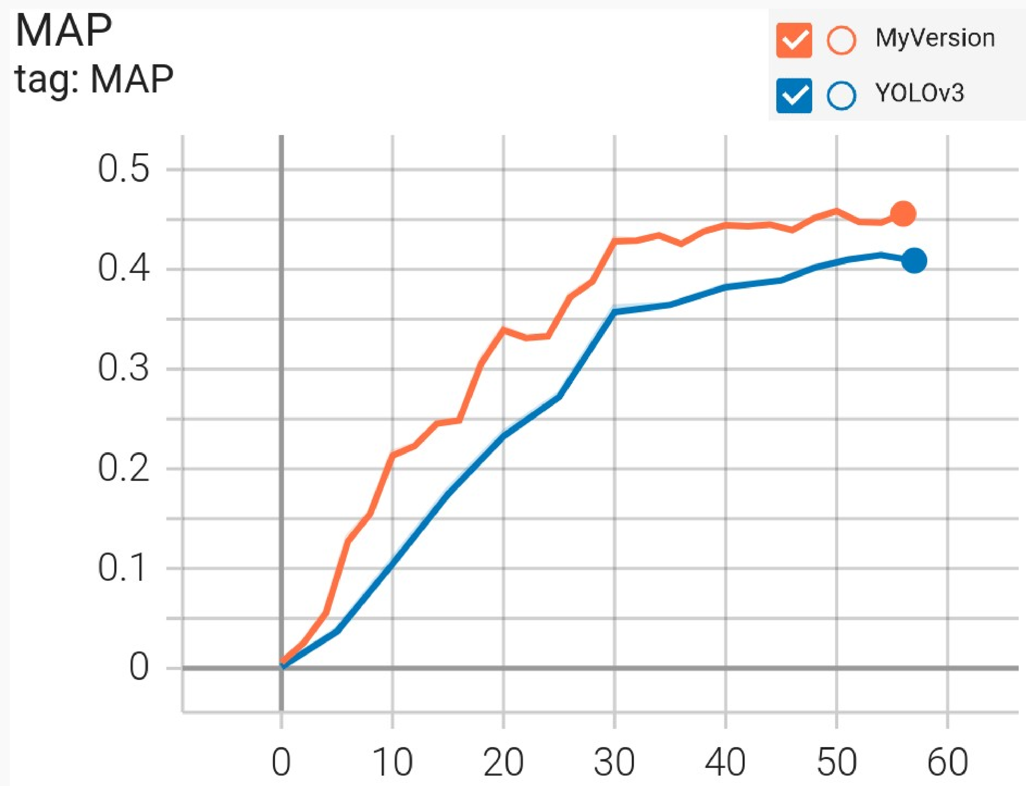
- The distance estimation is built on information from the object detection model.
- A model, where 4 coordinates and the square area of bounding box are taken as inputs, and the output is a prediction of the distance measured from the camera to the detected object.

Lane detection

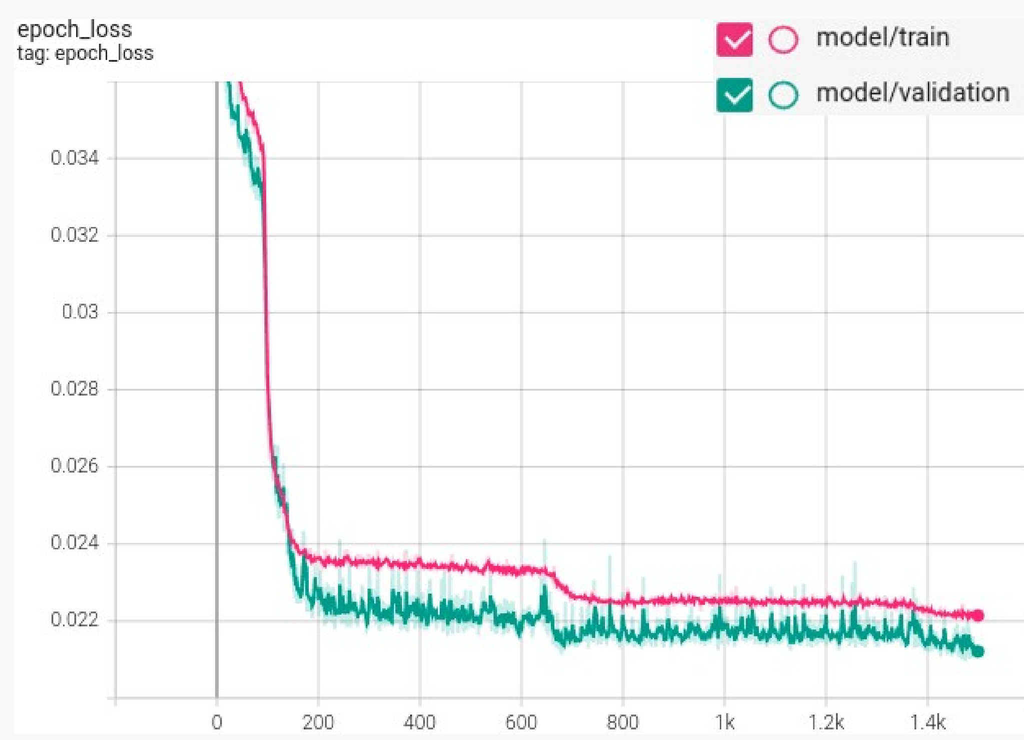
- The lane detection system is built on the basis of the OpenCV library, as it provides optimized algorithms, especially used for computer vision.
- The original frame is converted to grayscale and Gaussian blurring is used.
- Canny edge detection is applied, and the region of interest is selected such that most of the unwanted edges are left out.
- These edges are used in HoughLinesP algorithm. The output of HoughLinesP is the endpoints of the detected straight lines (x0, y0, x1, y1).
- Detected straight lines are grouped together by having negative or positive slope. The average position of each group represents the final line.

Results of the Computation

For almost 60 epochs, the training part went well. Where the custom model showed faster convergency than the original model and mAP of 0.445 was reached. Then further training became problematic, as both models stopped making better results.
For distance estimation, MSE = 0.021 was reached for the test dataset.

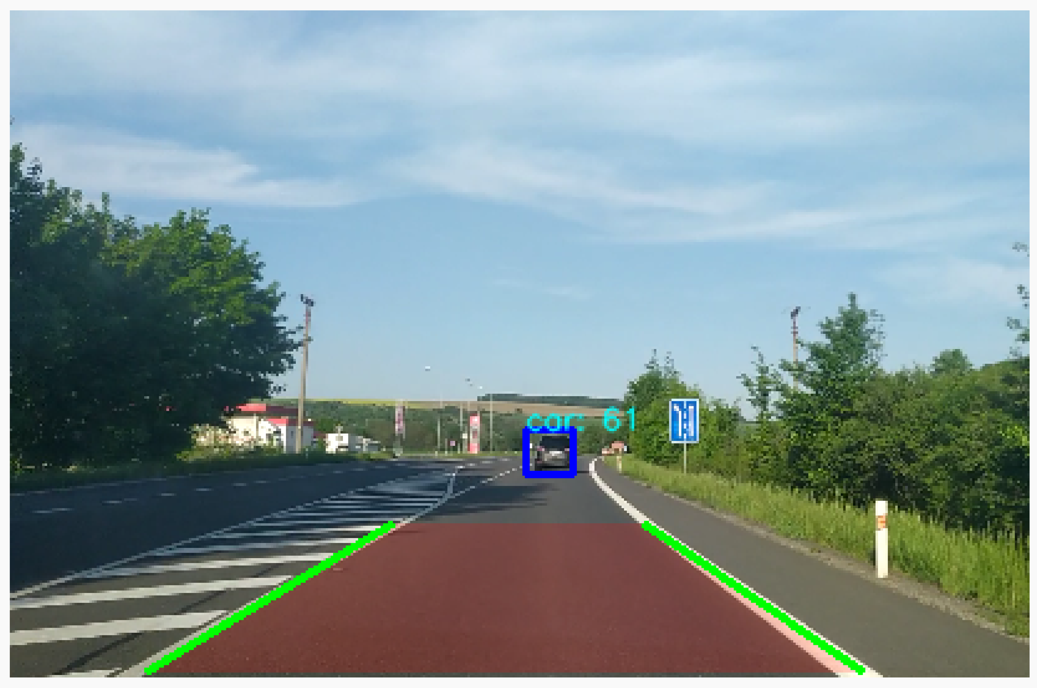


(a) Object detection model comparison



(b) Distance estimation model training

Figure: Object detection and distance estimation models training and evaluation



(a) Final result during good conditions

(b) Final result during bad conditions

Figure: Custom computer vision system for an autonomous system

Conclusion

- The whole system performs quite well under certain conditions. However, the system had a much higher number of false positive boxes (FPs) when the images included bad weather conditions, poor illumination, or a lot of shadows.

References

Gibson, P., Cano, J., Turner, J., Crowley, E. J., O'Boyle, M. & Storkey, A. (2020), 'Optimizing grouped convolutions on edge devices'.
URL: *https://arxiv.org/abs/2006.09791*

Jain, H. & Nandy, S. (2019), 'Incremental training for image classification of unseen objects'.

Janai, J., Güney, F., Behl, A. & Geiger, A. (n.d.), *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art*, Max-Planck-Institute for Intelligent Systems Tübingen and University of Tübingen, Germany.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* F. Pereira, C. Burges, L. Bottou & K. Weinberger, eds, 'Advances in Neural Information Processing Systems', Vol. 25, Curran Associates, Inc.
URL: *https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf*

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2015), 'You only look once: Unified, real-time object detection'.
URL: *https://arxiv.org/abs/1506.02640*

Zhang, C. & Zhang, Z. (2010), *A Survey of Recent Advances in Face Detection*.

Zou, Z., Shi, Z., Guo, Y. & Ye, J. (2019), 'Object detection in 20 years: A survey'.
URL: *https://arxiv.org/abs/1905.05055*