Deep Convolutional Neural Networks for Semantic Scene Parsing Individual Report

Wesley Quispel *4014901*

wesley.quispel@gmail.com

Introduction

The problem for our project was traffic scene segmentation, where we tried to classify traffic scene images from various locations in Europe into various classes. Scene segmentation is currently a large problem for the automotive industry to solve. It is too expensive to compute real-time, when given a datafeed at a reasonable frame rate and improvements in performance while retaining a very low classification error rate are essential to the adoption of autonomous driving.

Approach

Therefore our project group set out to build a network that would be faster to run while hopefully retaining the relative good performance of the reference model called SegNet-Basic.

The dataset we trained on was the Cityscapes dataset, which contained 30 classes divided into 8 super classes. To try to improve the performance and accuracy we divided the entire dataset into only 5 classes. The reasoning behind this was that it would be easier to classify a class correctly because most of the classes were closely related and bundling them could potentially mean that we would be able to reduce the amount of feature maps and layers we needed to use.

While the original SegNet network used 4 layers and 64 feature maps, we reduced this to 3 and 32 respectively so we could take advantage of the amount of the reduced amount of classes.

This network was then trained with Caffe on CPU mode. Even though this was much slower than GPU-mode it was easier to install and our test systems did not have the proper support for GPU mode. We also requested the use of the cluster but in the end this proved to cumbersome to get it to work. It would have been easier to test multiple versions if we had used the GPU mode of Caffe.

After training about 60% was correctly classified in to the 5 classes in the final version. This is not a marked improvement on the SegNet-Basic version and moreover the most crucial class “Do-not-hit” had a high classification error rate of around 10%.

Conclusions

Reducing the amount of layers in the architecture has yielded the same accuracy for a lower amount of classes. However reducing the size of the network was perhaps not the best way to train the network even though this was in line of our project to try to improve inference speed.

Reduction of amount of classes is also perhaps not the best way to improve accuracy and performance, since the in-class variation is high, as in objects like a car and a bicycle do not necessarily share the same feature maps so to reduce the amount of feature maps is not beneficial for the accuracy of the model. Increasing the amount of feature maps from 32 to say 64 (same as SegNet) or higher could help improve IoU, at the cost of the execution time of the network.

Performance should not come at the cost of lower accuracy when lives are at stake. In short, I would not drive in a car with our current model.