Free-Space Detection by Transfer Learning

Yu-Wei Lin
Stanford University
yuweilin@stanford.edu

Abstract

Extracting image features via different neural networks, like alexnet and googlenet, and using the features to do pixel-wise classification to determine free-space on road.

Figure 1. Training Image 01. Blue area is the correct prediction, red area is the false negative, and green area is the false positive.

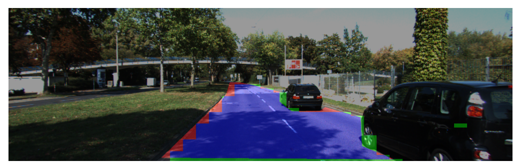

Figure 2. Training Image 02.

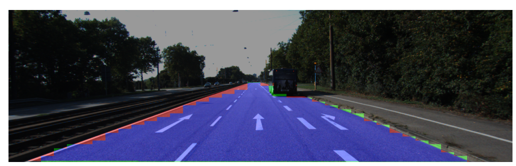

Figure 3. Training Image 03.

1. Introduction

To provide more assistance in driving, or even allowing vehicles to drive itself, determining the free space on road is essential. In this project, I would like to use different models/networks[4][1] provided in Caffe Model Zoo for features extraction and do pixel-wise classification to achieve the goal. The dataset I used is the open dataset (KITTI) provided by KIT [2]. It contains images taken from the front camera and labels for free space.

This topic is related to scene parsing in computer vision, and there are several impressive works over the years. One of the most impressed methods when I found is using 2 convolutional layers and followed by four layers of deconvolutional layers [3], which I would like to try out after this.

2. Approach

The approach I tried is pretty straight forward: passing images to a network to extract features, resize the features, and train the classifier. The only tricky part is since there are pixels that are always 'not-road' in the KITTI data, so I were not able to use the features directly but resize it to match the original image. However, since both networks I tried have several convolutional layers, making the extracted features mush smaller than original image, I tried to resize both original images and features, which means I were doing 'region-wise' classification instead of pixel-wise.

3. Conclusion

The pixel-wise accuracy for training and validation are both over 70%. The figures are some samples from the training and validation set. The blue area is the correct prediction, red area is the false negative, and green area is the false positive. We can see that most of the time classifier is able to distinguish road from other objects, like sky, cars, signs, walls, trees, etc. However, since I were actually doing 'region-wise' classification, we can see the un-smoothness on the edge. Also, as shown in Validation Image 2 5, which is the worst among all validation images, when the classifier saw the unseen pattern (white lines), it can be easily confused.

I would like to try the [3] approach for the next step, since it was reported to have a smoother and high accuracy result.

References

[1] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. pages 2155–2162, 2014.

Figure 4. Validation Image 01.

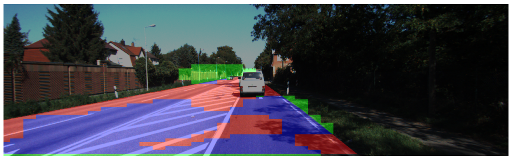

Figure 5. Validation Image 02.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[3] R. Mohan. Deep deconvolutional networks for scene parsing. *arXiv preprint arXiv:1411.4101*, 2014.

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.