Object Detection for the Audi Autonomous Driving Cup

Felix Wagner, Christoph Lehmann, and Klaus Dorer

Hochschule Offenburg
{felix.wagner/christoph.lehmann/klaus.dorer}@hs-offenburg.de

Abstract. One of the challenges for autonomous driving in general is to detect objects in the car's camera images. In the Audi Autonomous Driving Cup (AADC)[1], among those objects are other cars, adult and child pedestrians and emergency vehicle lighting. We show that with recent deep learning networks we are able to detect these objects reliably on the limited Hardware of the model cars. Also, the same deep network is used to detect road features like mid lines, stop lines and even complete crossings. Best results are achieved using Faster R-CNN with Inception v2 showing an overall accuracy of 0.84 at 7 Hz.

Keywords: object detection, deep learning, autonomous driving

1 Introduction

Since 2015, the German car manufacturer Audi runs a competition in the area of autonomous driving, in which student teams compete in autonomous driving challenges. One of the challenges is to detect objects in the car's camera images while driving. Among those objects are other cars, adult and child pedestrians for which barbie dolls are used to fit to the scale of the cars and, new in 2018, blue emergency vehicle lighting. The cars are instructed to drive significantly slower, if a child is in the proximity of the road. They have to stop if a pedestrian wants to cross at a pedestrian crossing. Also, the cars have to drive to the right side of the street and stop if they detect an emergency car with flashing emergency light, or give way to them on crossings.

To detect these objects, the cars are equipped with a Basler daA1280-54uc wide angle front camera (170 °) with a resolution of 1280 x 960px at 45Hz. A GeForce GTX 1050Ti on board can be used for image processing. The ceiling and bonnet in the images are cut off in this work to a final resolution of 1240 x 330px.

[1] https://www.audi-autonomous-driving-cup.com

2 Object Detection

To determine the best solution for object detection, several deep learning techniques were tested. There are two kinds of approaches for object detection. Candidates for one stage approaches are SSD and YoloV3. Two stage approaches are R-FCN, R-CNN and it's optimizations Fast R-CNN and Faster R-CNN.

The Single Shot MultiBox Detector (SSD) is based on the VGG-16 network [1] which is slightly modified. The architecture can be explained in three parts by its name. Single Shot describes the task of object localization and classification in a single forward pass through the network. MultiBox is the technique to determine the bounding boxes. The Detector detects objects and does a classification of those [2].

The basic approach of detection in the You Only Look Once (YOLO) architecture is based on a fully convolutional neural network which predicts the bounding boxes and classification of the image in one stage [3]. Version 3 of this architecture (YoloV3) is an optimization with the goal to reach a better accuracy sacrificing performance. The improved accuracy is based on three different sized detection kernels in the network instead of one [4].

The R-CNN (Region-based Convolutional Network) is a two stage object detection architecture. In stage one, the image is separated in 2000 region proposals. For every region, a convolutional neural network extracts a 4096-dimensional feature vector. In stage two, a Support Vector Machine (SVM) is used to calculate the probability of a searched object in each feature vector [5]. Fast R-CNN is an improvement of R-CNN. The CNN is now getting the input image instead of region proposals and generating a convolutional feature map. In the next step, a RoI pooling layer extracts the regions, which are finally passed to fully connected layers for classification and bounding box regression [6]. Both approaches are based on selective search to find the region proposals. This time-consuming process is eliminated in the Faster R-CNN architecture. Instead of selective search, a separate network is used to learn and predict the regions [7].

The Region-based Fully Convolutional Network (R-FCN) works similarly to the R-CNN approaches in finding the region proposals. The main difference is the removed fully connected layer after the RoI pooling. Score maps are generated before the RoI Pooling layer. After the pooling, classification and localization are based on average voting which results in a faster architecture than Faster R-CNN [8].

Every mentioned object detector has an underlying feature extractor, a deep neural network. There are several architectures of networks which can be used. The ResNet architecture is characterized by the fact that

the layers are not only feeding the following layer, but each layer also has connections to layers with 2 to 3 hops distance. Blocks with this property are called residual blocks. Due to the fact that the optimal number of layers required in the network isn't known, this method results in the ability of the network to skip layers during training that do not add value to the overall accuracy [9].

Another architecture is called Inception. One problem in designing of neural networks is choosing the right kernel size. Instead of making the network deeper to fit this problem, the inception architecture uses filters of multiple sizes on the same level and makes the network wider [10]. Inception v2 is intended to upgrade the accuracy and reduce the computational complexity by avoiding to alter the dimensions of the input and using smart factorization methods [11].

3 Results

A set of 15.343 manually labeled images or synthetically generated and labeled images [12] were used to train the networks pre-trained on coco[2] or kitti[3]. The images contain 38791 objects with a class distribution shown in Figure 1.

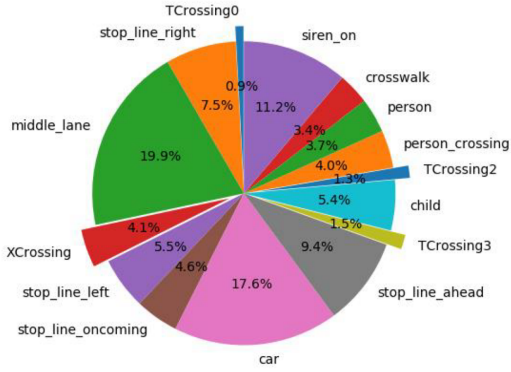

Fig. 1. Distribution of input images by class [13]

Table 1 shows the results of the combinations of region proposal algorithm and feature extractor evaluated. The best precision is achieved

[2] http://cocodataset.org
[3] http://www.cvlibs.net/datasets/kitti/

using R-FCN with ResNet101 achieving 0.86 mean average precision on at least 0.5 intersection over union (mAP@0.5IoU). At the slow speeds the model cars are driving, the 4.8 fps are acceptable. The best result due to its higher performance of 7 fps with an almost equal precision of 0.84 was achieved using Faster R-CNN with Inception v2. An example image is shown in Figure 2.

Table 1. Results of different deep learning networks [13]

Detector	Feature Extractor	max proposals	mAP@0.5IoU	FPS
Faster R-CNN	ResNet101	300	0.84	2.1
Faster R-CNN	ResNet101	200	0.83	2.6
Faster R-CNN	ResNet101	100	0.70	3.8
R-FCN	ResNet101	-	0.86	4.8
Faster R-CNN	ResNet50	300	0.84	3.7
Faster R-CNN	Inception v2	-	0.84	7.0

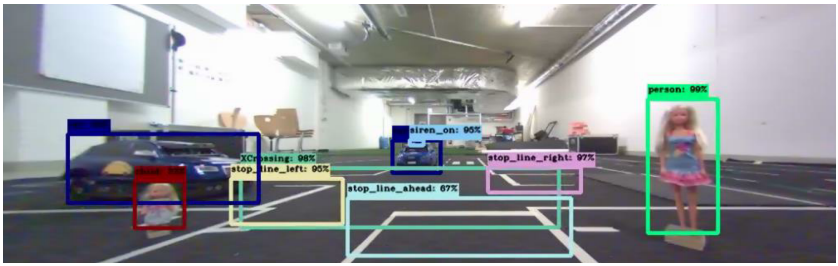

Fig. 2. Example detections with Faster R-CNN on Inception V2 [13]

4 Conclusions

In this paper we have shown that recent deep networks can run on the Audi Autonomous Driving Cup cars' GTX 1050Ti with reasonable frame and detection rates. This is remarkable given that 15 different classes of objects have been trained into the network.

A considerably higher accuracy can be achieved when reducing the number of classes to detect. Running a network with full crossing detection only achieves, for example, an accuracy of 0.987 despite the complex

structure of X (see Figure 2) and T crossings. However, running multiple networks concurrently is not feasible on the single graphics card of the cars performance wise.

Currently, at higher speeds of the car, detection rates drop probably due to motion blur. This is not a general limitation of the deep learning networks, but indicates that more training images are required at higher speeds of the car.

References

1. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
2. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: Computer Vision – ECCV 2016, Cham, Springer International Publishing (2016) 21–37
3. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. arXiv:1506.02640 (2015)
4. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv:1804.02767 (2018)
5. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 (2013)
6. Girshick, R.B.: Fast R-CNN. arXiv:1504.08083 (2015)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 91–99
8. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 379–387
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv:1512.03385 (2015)
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. arXiv:1409.4842 (2014)
11. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv:1512.00567 (2015)
12. Isenmann, R.: Evaluation von Deep Learning Verfahren mittels synthetisch generierter Bilder fuer autonomes Fahren. Master's thesis, Hochschule Offenburg, Germany (2018)
13. Lehmann, C.: Erkennung von Kreuzungen aus Videobildern eines autonomen Fahrzeuges mit Hilfe von Deep Learning. Master's thesis, Hochschule Offenburg, Germany (2019)