Deep Learning Based 3D Object Detection for Automotive Radar and Camera

Michael Meyer[*], Georg Kuschk[*]
Astyx GmbH, Germany
{m.meyer, g.kuschk}@astyx.de

***Abstract*— In this paper it is demonstrated how 3D object detection can be achieved using deep learning on radar pointclouds and camera images. A deep convolutional neural network is trained with manually labelled bounding boxes to detect cars. The results are compared to a deep neural network trained on lidar pointclouds and camera images. The average precision (AP) is used to evaluate the performance. For radar and camera the AP is 0.45, 0.48, and 0.61, whereas the AP for lidar and camera is 0.33, 0.35, and 0.46 for occluded, partially occluded and not occluded cars respectively. The performance of the network is significantly better with radar data compared to lidar data. Currently, the main limitation of the performance of the object detection with radar data and camera images is the dataset, which is until now rather small. However, the results show that deep learning is generally a suitable method for object detection on radar data.**

***Keywords*— object detection, machine learning, neural network, sensor fusion, radar, camera**

I. Introduction

One of the main perception tasks of autonomous vehicles is the detection of objects in their surroundings. Functional safety makes it desirable to have multiple complementary and redundant sensors which perform the object detection simultaneously. The most common sensors used for advanced driver assistance systems today are camera, lidar, and radar.

For camera images deep learning has become the state of the art method for 2D object detection [1], [2], [3]. It has also been shown that it is a suitable method for 3D object detection in lidar pointclouds [4], [5]. However, research has just begun to explore the possibilities to apply deep learning techniques on radar data. Recent publications used deep neural network for the following tasks: classification of range-doppler images [6], classification of objects in radar grid maps [7], and semantic segmentation of radar pointclouds [8].

When a convolutional neural network is used to classify range-doppler images, it is not able to differentiate between multiple objects in one image and is also not able to localize the objects. Lombacher *et al.* [7] classified each cell in a grid map, which means they segmented the radar grid into the classes *car* and *no-car*. This provides information about the location of the objects, but does not give information about the object instance the grid cell belongs to. Thus, it is not possible to infer information about neither the number of objects, nor the extent of each object.

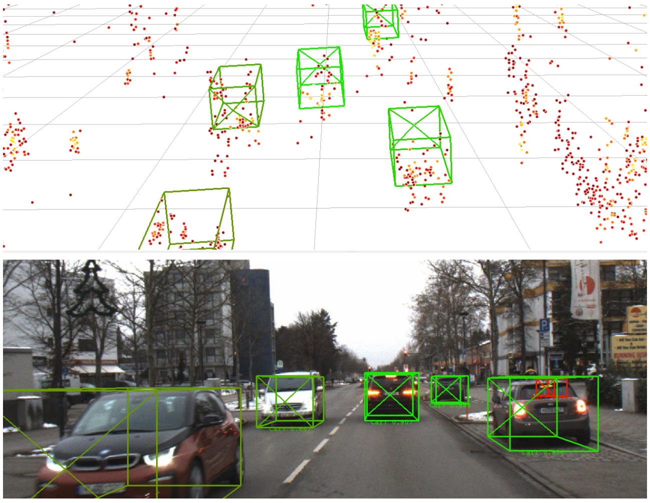


Fig. 1. Example results of 3D detections on test data. The color of the points represents the magnitude. The color of the detected boxes represents the score of the detection.

Many existing algorithms have been developed for 3D object detection based on lidar and camera (e.g. [9], [10], [11]). Since some radar sensors, like lidar sensors, output 3D pointclouds (even though the pointcloud has completely different properties), one can apply similar networks on radar and camera data.

Lidar sensors are spatially very accurate, but prone to weather conditions like rain, fog, snow, or dust. In comparison, radar sensors are more reliable and robust but also more noisy and less accurate. So, it is not clear if network achitectures similar to the ones used for lidar are suitable for radar data.

At the same time radar-camera fusion is desirable because the sensors have very different characteristics and thus complement each other [12].

Until recently the output of radar sensors was too sparse to apply deep learning methods for radar-camera fusion on the raw data level. To the best of the authors' knowledge there have not been any publications about low level radar camera fusion with deep learning based Convolutional Neural Networks (CNNs).

In this paper a deep neural network is used to do 3D object detection on a radar pointcloud and a camera image. The network is similar to the network used by Ku *et al.* [11].

[*] Both authors contributed equally to this paper.

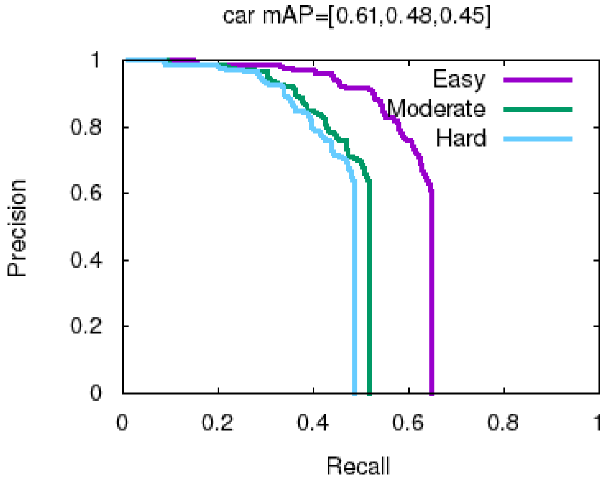


Fig. 2. Precision - recall curve for radar and camera on test dataset.

The performance of the network is evaluated and radar-camera vs. lidar-camera performance compared.

II. Methods

A CNN is trained to detect cars in 3D space based on a pointcloud and an RGB image as input. Each pointcloud and image belong to synchronized ”frames” of a dataset containing public roads recorded in the south of Germany [13]. The training is performed on one split of the dataset and the evaluation is performed on another split, such that data used for training is not used for evaluation. Training and evaluation are done once with radar pointcloud plus camera images and once with lidar pointcloud plus camera images.

A. Network

A pointcloud consisting of N points $(x, y, z,$ Magnitude) and a RGB image is used as input for a network which then outputs bounding box predictions with position and dimension in 3D space based on the input. The network achitecture used in this paper is similar to the network described in [11] (see Fig. 3). However, in contrast to the paper no *Feature Pyramid Network* (FPN) is used in the feature extractor. Since in this paper the network is only trained to detect the class *Car* with relatively large objects, the FPN is not so relevant.

The pointcloud of each frame is used to generate a bird eye view (BEV) image with six height maps and one density map. The doppler information of the radar data is not used. A 3D region proposal network (based on an adaptation of the network *VGG* [14]) is then used to generate proposals based on camera images and BEV images of the radar pointcloud.

The proposals are used to predict boxes which are encoded through four corners and two heights (offsets from ground to bottom and top of the box). This rather unintuitive 14-dimensional representation is used because it has been shown to yield the best results [11]. Additionally, an angle is predicted to determine which side between the four corners belongs to the front of the detected object.

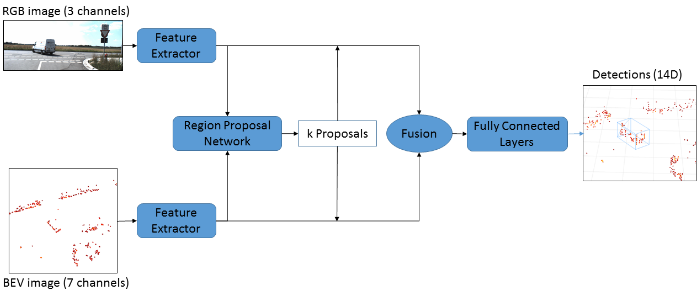


Fig. 3. Scheme of the architecture of the network. The BEV image consists of 6 height maps and one density map. The detections are encoded as 4 corners and two height offsets from the ground.

B. Dataset

For training and evaluation a dataset containing 455 frames of synchronized camera, lidar, and radar data is used [13]. Each radar pointcloud contains approximately 1000 - 10000 points, obtained using the Astyx 6455 HiRes sensor. Each point contains x,y,z position, magnitude, and doppler information (radial velocity). The camera images have a size of 2048×618 pixels and were captured with a Point Grey Blackfly camera. The lidar pointcloud was obtained with a Velodyne VLP-16.

C. Training

The data is split into a training and a test set using a ratio of 4:1. For deep learning applications this dataset is very small. Therefore, two data augmentation methods are used during training. The first method is flipping the image, the pointcloud, and the groundtruth boxes horizontally. The second method is adding noise to the camera image with a PCA based method, which is known as *Fancy Principle Component Analysis* and was introduced by Krizhevsky in his famous AlexNet paper [15].

The network is trained for 22000 iterations with a learning rate of 0.0001 and a mini batch size of 16. The training took 3 hours and 24 minutes on two Nvidia Titan V GPUs.

III. Results

The training is evaluated on the test dataset. One exemplary frame from the test dataset is displayed with the detections from the network in Fig. 1. For evaluation the groundtruth is split into three categories *Easy*, *Moderate*, and *Hard*. For the latter all objects are evaluated, whereas for *Moderate* fully occluded objects are excluded, and for *Easy* only fully visible objects are evaluated.

The average precision (AP) which is a well established evaluation metric for 3D object detection evaluation [16] is utilized. Detections are matched to groundtruth objects when they have a 3D intersection over union (IoU) above 0.5.

For radar-camera the precision-recall curve is displayed in Fig. 2. For the class *Car* the average precision is calculated to be 0.61, 0.48, 0.45 for *Easy*, *Moderate*, and *Hard* respectively.

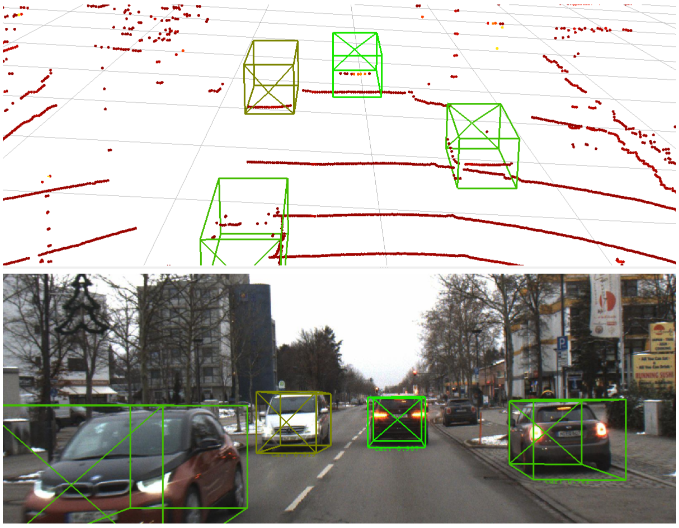


Fig. 4. Example detections on lidar and camera data.

To have a meaningful comparison between radar and lidar, the performance of the network trained with radar and camera data is compared to its performance when it is trained with lidar and camera data of the same small dataset. All parameters of the training remain the same. The precision-recall curve of the resulting detections is displayed in Fig. 5. The average precision is 0.46, 0.35, 0.33 for *Easy*, *Moderate*, and *Hard* respectively. This means that the AP of the network is significantly higher when radar data is used instead of lidar pointclouds.

The orientations of the detections are plotted in a circular bar plot for radar plus camera and lidar plus camera in Fig. 7. It can be noted that by far the most detections have an orientation of 0° or 180°. This effect is more apparent in the detections based on radar.

IV. Discussion

The AP of the network with radar pointcloud and camera images is decent considering the small dataset. Obviously, it is considerably lower than state of the art results of object detection algorithms trained on lidar and camera. At the moment, the best performance reported on the KITTI 3D object detection benchmark [16] trained on lidar and camera is 0.68 for the category *Hard*. But the KITTI dataset contains nearly 20 times the amount of training data and therefore these results are not really comparable. With the small dataset used in this paper the results are significantly better when radar is used.

Lidar gives a spatially very accurate pointcloud. Thus, it might come as a surprise that the object detection performance is so much better when radar data is used as input instead of a lidar pointcloud. Additionally, it would have been expected that when radar data is used the network is relatively better in detecting occluded objects. On non-occlud objects the performance is 15% better with radar while on occluded objects it is only 12% better. So, even though the network is indeed better at detecting occluded objects, relative to fully visible objects the network is not performing better with radar pointclouds. This can be explained by the fact that even though lidar is more spatially accurate than radar, the density of the pointcloud is not evenly distributed. The pointcloud is very dense close to the lidar sensor and very sparse further away. Thus, far away objects do not always have lidar points even when they are not occluded. Without points in the pointcloud belonging to the object it is very unlikely that the object is detected (see Fig. 6). These might be objects that are partially occluded, but it is also common that objects which are not occluded have no lidar points on their surface. Of course this effect can be mitigated by using a lidar sensor with more layers. Nevertheless, this would just shift the problem to a greater range, since the uneven distribution of points is a defect inherent to the characteristics of a lidar sensor.

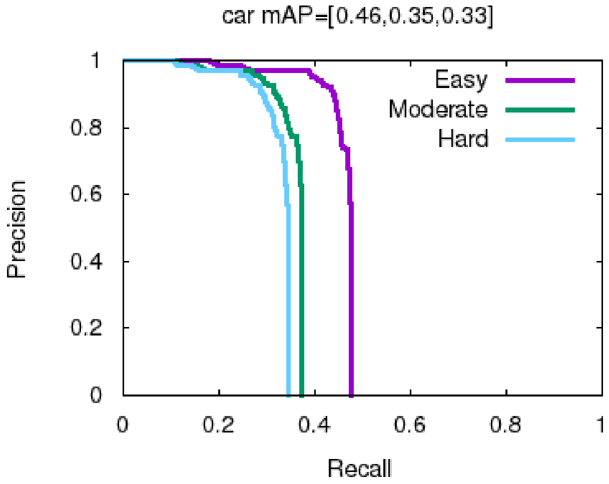


Fig. 5. Precision - recall curve for lidar and camera on the test dataset.

For radar pointcloud and camera images the average precision is surprisingly high for such a small dataset. From the AP value on the test dataset alone, one could conclude that the amount of training data is nearly sufficient for the network to generalize. However, once the orientations of the detections on the test data are considered it becomes evident that the variance of orientations in either the training data or test data (or both) is not sufficient. Nearly all detection boxes on the test dataset are parallel to the view axes of the ego vehicle. This would be desirable if the groundtruth of these frames indeed had these orientations. Unfortunately, this is not the case. Training the network to detect cars was successful, but the network is not yet able to accurately predict the corners of the vehicles. Based on these results, it can be suspected that increasing the dataset size and using training data in which the oriantations are more evenly distributed will result in significantly better results.

It is difficult to conclude why the AP of the network is better with radar while the orientation of the detections seem worse. One possibility is that it is just caused by the fact that the lidar data used in this paper was from a 16-layer lidar sensor whose layers missed a lot of objects. Another explanation is that cars appear more generic in the radar

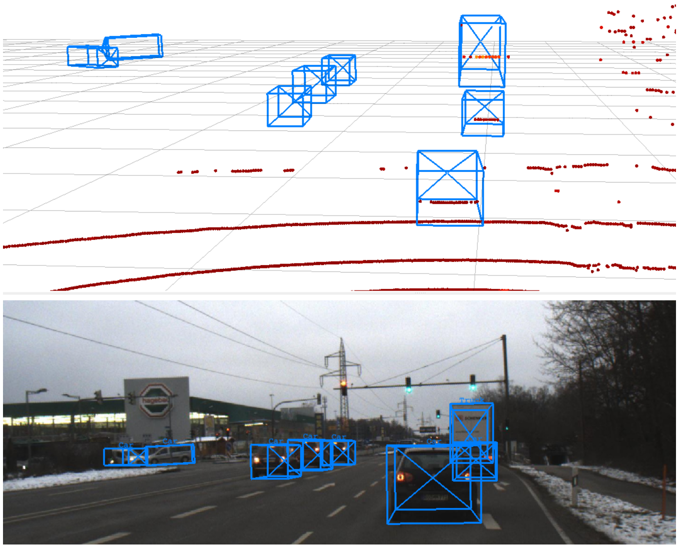


Fig. 6. Example frame of groundtruth that shows the main shortcoming of the lidar sensor. All five cars on the left have no lidar points on their surface.

pointcloud regarding the angle of the vehicle. This means that the network is able to generalize easier on the radar data while being able to do better predictions of the orientation in the lidar pointcloud. Further studies are required to answer this question.

V. Conclusion and Future Work

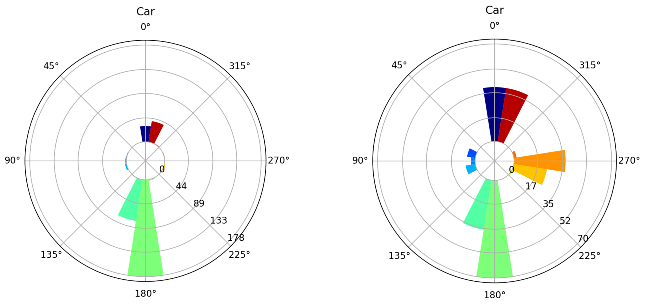
In this paper it has been shown that deep CNNs are a suitable method for low level sensor fusion of radar pointclouds and camera images. In the test dataset of radar pointclouds and camera images fully visible cars were detected with an average precision of 0.61.

Radar data is not as spatially accurate as lidar pointclouds. The results show that this is no problem for the object detection task. The network seems able to model the noise inherent to the radar data. This is no surprise since it has been shown that training neural networks with noisy data can be beneficial for the robustness of the performance [17].

Even though the doppler information, which can be assumed to be a powerful feature of the radar points, is not included in the input for the network, the results are quite acceptable. In future work the doppler information of the pointcloud will be included in the input of the network. Additionally, uncertainty will be predicted by network in order to improve the training process of the network [18].

Furthermore, object detection results based on radar and camera will be compared with results with camera and a 64-layer lidar trained with the same dataset.

The size of the dataset will also be increased, which will probably improve the performance significantly. The dataset size is assumed to be currently the most crucial limitation for the performance of 3D object detection on radar pointclouds and camera images in this paper.



(a) Radar and camera

(b) Lidar and camera

Fig. 7. Circular bar plot of the orientations of detection bounding boxes on test dataset.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, June 2018.

[5] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018.

[6] R. Perez, F. Schubert, R. Rasshofer, and E. Biebl, "Single-frame vulnerable road users classification with a 77 ghz fmcw radar sensor and a convolutional neural network," in *Intl. Radar Symposium (IRS)*, 2018.

[7] J. Lombacher, K. Laudt, M. Hahn, J. Dickmann, and C. Wohler, "Semantic radar grids," in *Intelligent Vehicles Symposium (IV)*, 2017.

[8] O. Schumann, M. Hahn, J. Dickmann, and C. Wohler, "Semantic segmentation on radar point clouds," in *Intl. Conference on Information Fusion (FUSION)*, 2018.

[9] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *CVPR*, 2017.

[10] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *CVPR*, June 2018.

[11] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *Intl. Conference on Intelligent Robots and Systems (IROS)*, 2018.

[12] Z. Zhong, S. Liu, M. Mathew, and A. Dubey, "Camera radar fusion for increased reliability in adas applications," *Electronic Imaging*, vol. 2018, pp. 1–4, 01 2018.

[13] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3d object detection," in *Manuscript submitted to EuRad 2019*.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[17] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise-enhanced convolutional neural networks," *Neural Networks*, vol. 78, pp. 15 – 23, 2016, special Issue on Neural Network Learning in Big Data.

[18] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.