Object Detection and Classification in Occupancy Grid Maps using Deep Convolutional Networks

Sascha Wirges, Tom Fischer and Christoph Stiller
Mobile Perception Systems Group
FZI Research Center for Information Technology
Karlsruhe, Germany
Email: {wirges,fischer,stiller}@fzi.de

Jesus Balado Frias
Applied Geotechnologies Research Group
University of Vigo
Vigo, Spain
Email: jbalado@uvigo.es

Abstract—A detailed environment perception is a crucial component of automated vehicles. However, to deal with the amount of perceived information, we also require segmentation strategies. Based on a grid map environment representation, well-suited for sensor fusion, free-space estimation and machine learning, we detect and classify objects using deep convolutional neural networks. As input for our networks we use a multi-layer grid map efficiently encoding 3D range sensor information. The inference output consists of a list of rotated bounding boxes with associated semantic classes. We conduct extensive ablation studies, highlight important design considerations when using grid maps and evaluate our models on the KITTI Bird's Eye View benchmark. Qualitative and quantitative benchmark results show that we achieve robust detection and state of the art accuracy solely using top-view grid maps from range sensor data.

I. INTRODUCTION

We require a detailed environment representation for the safe use of mobile robotic systems, e.g. in automated driving. To enable higher level scene understanding and decrease computational cost for existing methods, information needs to be further filtered, segmented and categorized. This task can be accomplished by object detection, shape estimation and classification methods, in the following simply referred to as *object detection*. Given an input environment representation, the object detector should output a list of oriented shapes and their corresponding most likely semantic classes.

In this work we represent the environment by top-view grid maps, in the following referred to as *grid maps*. Occupancy grid maps, first introduced in [1] encode surface point positions and free-space from a point of view in a two-dimensional grid. As all traffic participants move on a common ground surface one might not require full 3D information but instead represent the scene in 2D with obstacles occupying areas along the drivable path. Multi-layer grids are well-suited for sensor fusion [2] and their organized 2D representation enables the use of efficient convolutional operations for deep learning in contrast to sparse point sets. Whereas objects in camera images might vary in scale due to the projective mapping grid maps representing an
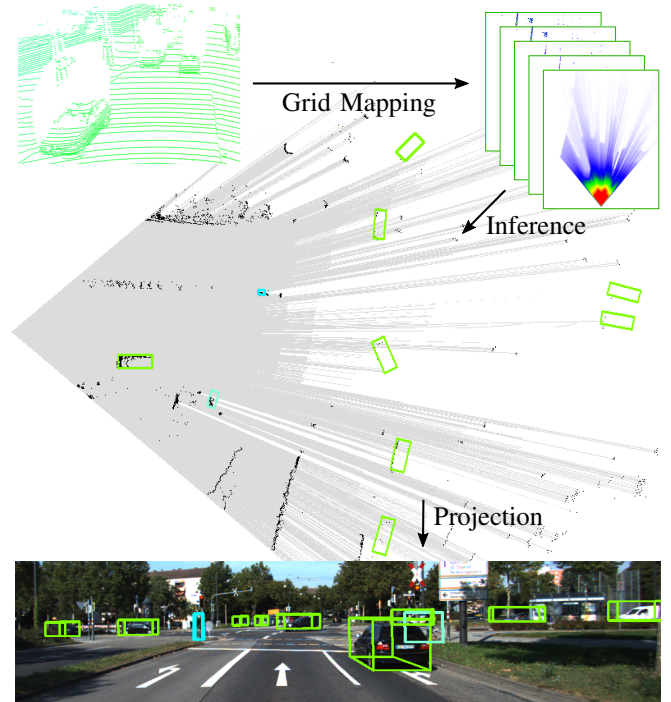
Figure 1: System overview. We transform range sensor measurements to a multi-layer grid map which serves as input for our object detection and classification network. From these top view grid maps the network infers rotated 3D bounding boxes together with semantic classes. These boxes can be projected into the camera image for visual validation. Cars are depicted green, cyclists aquamarin and pedestrians cyan.

orthographic top view are composed of metric fixed-size cells making objects scale invariant. In addition, object projections in camera images might overlap which is not the case for multiple objects in occupancy grid maps.

Here, we first present an overview on object detection and semantic classification in multi-layer grid maps. We then train different meta-architectures, show the influence of various parameters and discuss their effects on performance in detail. By making specific design considerations for the grid map

domain we are able to train object detectors in an end-to-end fashion achieving state-of-the-art accuracy at reasonable processing time compared to recent 3D object detection approaches. Finally, we compare the most promising object detection models to recent state-of-the-art approaches on the KITTI bird's eye view benchmark.

First, we review and compare related work on object detection in grid maps and other domains in Section II. We then present our preprocessing to obtain training examples in Section III. After recalling our general training strategy and metrics we provide information on the grid map domain adaptation in Section IV. We perform a quantitative and qualitative evaluation of different configurations in Section V. Finally, we conclude our work and propose future plans for object detection in Section VI.

II. RELATED WORK

A. Object Detection Meta-Architectures

Recently, a notable amount of state-of-the-art object detectors is based on the Faster R-CNN meta-architecture [3]. In Faster R-CNN detection happens in two stages, a region proposal network (RPN) and a classification and box refinement network. In the RPN features are extracted from the input and used to predict class-agnostic box candidates in a grid of anchors tiled in space, scale and aspect ratio. The feature slice corresponding to each box proposal is then sequentially fed into the box classifier. In the original Faster R-CNN implementation each feature slice is fed into two dense layers before performing classification and box refinement whereas in R-FCN [4] the dense layers are omitted, reducing the amount of computation per region. In contrast to Faster R-CNN and R-FCN single shot detectors (SSDs) [5] predict bounding boxes and semantic classes with a single feed-forward CNN, significantly reducing inference time but also lowering the overall accuracy.

B. Feature Extractors

The detection stage input consists of high-level features. These features may be computed by a deep feature extractor such as Resnet [6], Inception [7] or MobileNet [8]. Resnets implement layers as residual functions, gain accuracy from increased depth and were successfully applied in the ILSVRC and COCO 2015 challenges. Among other aspects, Inception and MobileNet use factorized convolutions to optimize accuracy and computation time. With Inception units the depth and width of networks can be increased without increasing computational cost. MobileNets further reduce the number of parameters by using depth-wise separable convolutions.

C. Object Detection in Aerial Images

Here, we compare the object segmentation task in grid maps to (scale-corrected) satellite or aerial images which has a long research history [9], [10], [11]. For example, [10] use 1420 labeled samples in high resolution panchromatic images to train a vehicle detector, reducing false positives by selecting only hypotheses on surfaces semantically classified as streets. Whereas atmospheric conditions might limit aerial

image quality due to camera views far from the scene top view grid maps suffer from occlusions due to a view within the scene. These problems can either be tackled by fusing multiple measurements from different views or learned environment reconstruction [12]. However, [13] consider the shadows / occlusions from cars one of the most relevant features (together with the rectangular shape and the windshield layout).

D. KITTI Bird's Eye View Benchmark

Training deep networks requires a comparably large amount of labeled data. The KITTI Bird's Eye View Evaluation 2017 [14] consists of 7481 training and 7518 camera images as well as corresponding range sensor data represented as point sets. Training and test data contain 80,256 labeled objects in total which are represented as oriented 3D bounding boxes (7 parameters). As summarized in Table I, there are eight semantic classes labeled in the training set although not all classes are used to determine the benchmark result.

Class	Occurrence %	Max. length m	Max. width m
Car	70.8	6.67	2.04
Pedestrian	11.1	1.44	1.20
Van	7.2	6.91	2.52
Cyclist	4.0	2.17	0.93
Truck	2.7	16.79	3.01
Misc	2.4	12.6	2.68
Tram	1.3	35.24	2.81
Sitting person	0.6	1.33	0.78

Table I: Semantic classes available in the KITTI Bird's Eye View Evaluation 2017. Occurrences and max. length / width are provided for the training set. In the evaluation vans are no car false positives and sitting persons are no pedestrian false positives.

Currently, successful benchmark submissions share a two-stage structure comprised of RPN and box refinement and classification network [15], [16]. They first extract features from sensor data, create axis aligned object proposals and perform classification and box regression on the best candidates. Whereas the region proposal in [15] is based only on a grid map, [16] also incorporate camera images to generate proposals. To further increase accuracy [16] train two separate networks for cars and pedestrians/cyclists, respectively.

E. Choice of Input Features

The choice of grid cell features varies heavily along different publications. [17], [18], [15] use the (normalized) number of detections and characteristics derived from detection reflectances. As the reduction of 3D range sensor information to 2D implies a loss of information features that encode height information might be relevant. [17] use the average height and an estimate of its standard deviation as features whereas [15] use four height values, equally sampled in the interval between the lowest and the highest point coordinate of each cell.

There are also higher level features possible. [19] use evidence measures for occupied and free cells, average velocity

and its auto-covariance matrix estimated by a particle filter. [17] estimate the standard deviations in the two principle horizontal directions whereas [18] estimate local planarity. However, as we aim to train object detectors in an end-to-end fashion we do not consider handcrafted features in this work. On the one hand, it seems sometimes arbitrary to us how certain features are picked and there is no evidence of gaining accuracy when using higher-level features in combination with the training of deep networks. On the other hand, higher-level features such as velocity estimates might not be available at all times.

F. Box Encoding

Similar to the feature encoding of grid cells there is a variety of different box encodings used in related work. [15] use eight 3D points (24 parameters) for box regression and recover the box orientation in direction of the longer box side. In contrast to this, [16] use four ground points and the height of the upper and lower box face, respectively (14 parameters). They explicitly regress the sine and cosine of orientation to handle angle wrapping and increase regression robustness. One encoding that needs the minimum amount of 2D box parameters (5) is presented in [20]. They represent boxes by two points and one height parameter (5 parameters).

III. GRID MAP PROCESSING

We perform minimal preprocessing in order to obtain occupancy grid maps. As there are labeled objects only in the camera image we remove all points that are not in the camera's field of view (see Figure 2). We then apply optional ground surface segmentation described in Section III-A and estimate different grid cell features summarized in Section III-B. The resulting multi-layer grid maps are of size $80\,\text{m} \times 80\,\text{m}$ and a cell size of either $10\,\text{cm}$ or $15\,\text{cm}$.

A. Ground Surface Segmentation

Recent approaches create top view images including all available range sensor points. However, it remains unclear if ground surface points significantly influence the object detection accuracy. Therefore, we optionally split ground from non-ground points. As we observed the ground to be flat in most of the scenarios we fit a ground plane to the representing point set. However, any other method for ground surface estimation can be used as well. For each scan, we perform nonlinear Least-Squares optimization [21] to find the optimal plane parameters

$$\mathbf{pl}^* = \arg\min_{\mathbf{pl}} \sum_{\mathbf{p} \in \mathcal{P}} \rho\left(\|\mathbf{e}\left(\mathbf{pl}, \mathbf{p}\right)\|^2\right)$$

which minimize the accumulated point-to-plane error for all points $\mathbf{p}$ of the point set where $\mathbf{e}\left(\mathbf{pl}, \mathbf{p}\right)$ denotes the distance vector between $\mathbf{p}$ and its plane projection point. The loss function ρ is chosen to be the Cauchy loss with a small scale (5 cm) to strictly robustify against outliers. We then remove all points from the point set with signed distance below $0.2\,\text{m}$ to the plane.

B. Grid Cell Features

We use the full point set or a non-ground subset to construct a multi-layer grid map containing different features. Inspired by other contributions (e.g. [15], [16]) we investigate if there is evidence for better convergence or accuracy by normalizing the number of detections per cell.

Exemplary, we follow the approach presented in [22] to estimate the decay rate

$$\lambda_i = \frac{H_i}{\sum_{j \in \mathcal{J}} d_i(j)}$$

for each cell i as the ratio of the number of detections H_i and the sum of distances $d_i(j)$ traveled through i for all rays $j \in \mathcal{J}$. We determine $\mathcal{J}$ and d_i by casting rays from the sensor origin to end points using the slab method proposed in [23]. In another configuration, we use the number of detections and observations per cell directly. To encode height information we use the minimum and maximum z coordinate of all points within a cell instead of splitting the z range into several intervals (e.g. as in [15], [16]). In all configurations we determine the average reflected energy, in the following termed as intensity. Figure 2 depicts the grid cell features presented. Table II summarizes the feature configurations used for evaluation.

Features Id	Configuration
F1	Intensity, min. / max. z coordinate, detections, observations
F2	Intensity, min. / max. z coordinate, decay rate
F3	Intensity, detections, observations
F1*	Same as **F1** but with ground surface removed

Table II: Evaluated feature configurations.

IV. TRAINING

Out of the total amount of training examples we use 2331 (31%) samples for internal evaluation, referred to as the validation set. As summarized in Table IV we train networks with several configurations, varying one parameter at a time. We pretrain each feature extractor with a learning rate of $1 \cdot 10^{-3}$ (Resnet) and $6 \cdot 10^{-4}$ (Inception) for 250k iterations with an grid cell size of 15cm. A few networks are compared against other methods by uploading inferred labels to the KITTI benchmark. Due to our limited computational resources we train all networks using SGD, batch normalization [24] and use the Momentum optimizer with a momentum of 0.9. Starting from the trained baseline networks we then train each configuration for another 200k iterations with the learning rate lowered by a factor of 2.

A. Box Encoding

As mentioned in Section II-F there are several box encodings used. We want to use as few parameters as possible because we assume this to be beneficial for box regression accuracy. However, while the orientation estimation might be more problematic we adapt the approach in [16] and estimate the orientation θ by two parameters $\sin(2\theta)$ and $\cos(2\theta)$, providing an explicit and smooth mapping within $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ (B1). To compare against other encodings we also represent

(a) Intensity (b) Height difference

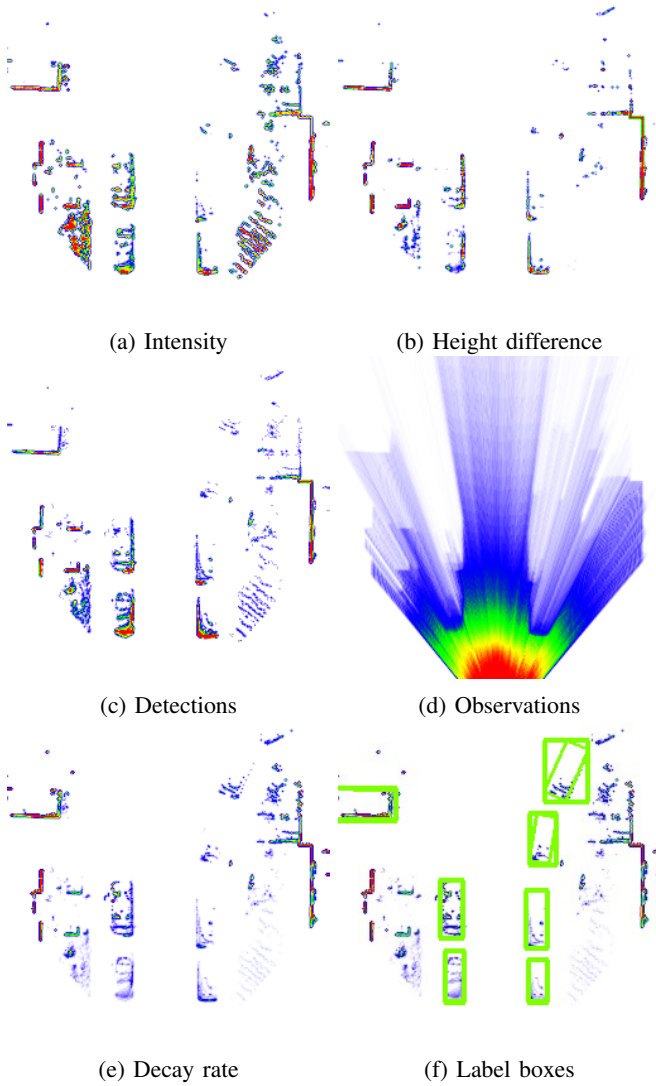(c) Detections (d) Observations

(e) Decay rate (f) Label boxes

Figure 2: Grid cell features (Fig. 2a–2e) and label boxes (Fig. 2f). Low values indicated by blue/white color, high values by red color. The intensity layer carries information on the surface material. The height difference layer (and consequently the min. / max. z coordinate layer) encodes information that would otherwise be lost due to the projection onto the ground surface. The number of detections depend on the object distance and their vertical structure while the number of observations describes the observable space. The decay rate (see [22]) is a normalized measure based on detections and observations. We use rotated and axis-aligned ground truth boxes with semantic classes (Fig. 2f). Best viewed digitally with zoom.

boxes by position, extent and orientation (B2) as well as two points and width [20] (B3). The encodings are summarized in Table III.

B. Data Augmentation

Because convolutional filters are not rotationally invariant we increase the amount of training samples by augmenting different viewing angles. Similar to [15] and [16] we

Box Encoding Id	Parameters
B1	$[x_c, y_c, w, h, \sin(2\theta), \cos(2\theta)]$
B2	$[x_c, y_c, w, h, \theta]$
B3	$[x_1, y_1, x_2, y_2, w]$

Table III: Evaluated box encodings.

randomly flip the grid map around its x-axis (pointing to the front). Subsequently, we randomly rotate each grid map within $[-15°, 15°]$ around the sensor origin. Label boxes are augmented accordingly.

C. Proposal Generation

In contrast to [16] we aim to train one network for many classes (see Table I). However, as vans and cars as well as sitting persons and pedestrians are similar or only very few training samples are available we merge these classes into one class.

Working on fixed scale grid maps, we can further adapt the object proposal generation to our domain by adapting its size, aspect ratio and stride. Table I summarizes the maximum length and width for each semantic class. Therefore, we determine a small set

$$\mathcal{S} = \{1.75\,m, 2.5\,m, 9\,m, 22\,m\}$$

of anchor sizes that enclose most objects closely. Note that we determine the combined extent for cars / vans and pedestrians / sitting persons as we treat them to be of the same class. Trams might not fit completely into the largest feature maps. However, we think that they can be distinguished properly due to their large size. We chose the feature slice aspect ratios to be 1:1, 2:1 and 1:2 and the stride to be 16 times the grid cell size.

D. Metrics

To train the RPN we use the same multi-task loss as presented in [3]. However, for the box classification and regression stage we extend this metric by another sibling output layer and define the multi-task loss similar to [20] as

$$L = L_{\text{cls}}(p, p^*) + \lambda_1 L_{\text{loc},1}(v, v^*) + \lambda_2 L_{\text{loc},2}(u, u^*).$$

For each proposal a discrete probability distribution p over $K + 1$ classes is computed by the softmax function. Here, $L_{\text{cls}}(p, p^*)$ denotes the multi-class cross entropy loss for the true class p^*. v is the predicted bounding-box regression offset given in [3] in which v specifies a scale-invariant translation and log-space height / width shift relative to an object proposal. u denotes the predicted inclined bounding-box regression offset. For the localization losses $L_{\text{loc},1}$ and $L_{\text{loc},2}$ we use the robust smooth L_1 loss. Here, v^* denotes the true bounding-box regression target and u^* the true inclined bounding-box regression target depending on the used box encoding (see Table III). The hyperparameters λ_1 and λ_2 balance the different loss terms and are set to $\lambda_1 = \lambda_2 = 2$ in all experiments. The difference between the two bounding box representations is also depicted in Figure 2f.

Net Id	Architecture Meta Arch.	Feat. Extr.	Grid Map Feat. Id	Res. cm	Box Enc. Id	Cars E	M	H	Cyclists E	M	H	Pedestrians E	M	H	Time ms
1	F. R-CNN	Res101	F1	15	B1	73.0	62.6	62.6	19.8	14.7	14.2	15.1	15.1	15.1	121
2	**R-FCN**	Res101	F1	15	B1	84.6	67.8	67.4	33.7	30.1	29.8	12.4	12.9	12.8	109
3	F. R-CNN	**IncV2**	F1	15	B1	60.9	53.3	53.2	29.7	27.4	22.7	13.0	12.4	12.3	113
4	F. R-CNN	Res101	**F2**	15	B1	82.7	72.8	66.5	26.9	24.7	20.3	10.8	11.7	11.6	121
5	F. R-CNN	Res101	F1	**10**	B1	87.5	76.5	75.6	39.5	40.0	35.2	35.4	36.2	36.5	272
6	F. R-CNN	Res101	F1	15	**B2**	31.1	27.3	27.8	2.0	9.1	9.1	2.5	2.9	2.9	125
7	F. R-CNN	Res101	**F3**	15	B1	64.2	51.8	52.0	18.8	15.8	16.0	5.3	5.1	4.8	119
8	F. R-CNN	Res101	**F1***	15	B1	73.1	63.8	63.0	13.8	12.0	11.7	10.5	10.4	10.5	123
4	F. R-CNN	Res101	**F2**	15	B1	80.2	66.4	64.7	40.8	28.1	27.8	14.2	11.7	12.8	121
5	F. R-CNN	Res101	F1	**10**	B1	86.1	75.4	67.6	51.8	37.6	37.4	28.8	24.5	20.1	272

Table IV: Object detection configurations with KITTI evaluation results on the validation set (upper part) and on the test set (lower part). Given the baseline configuration 1, we vary different parameters, one at a time.

V. EVALUATION

Table IV summarizes the evaluation results on the validation set for different network configurations.

A. Metrics

We evaluate the overall accuracy based on the average precision for the KITTI Bird's Eye View Evaluation using an Intersection over Union (IoU) threshold of 0.7 for cars and an IoU of 0.5 for cyclists and pedestrians. The evaluation is divided into the three difficulties Easy (E), Moderate (M) and Hard (H) based on occlusion level, maximal truncation and minimum bounding box size.

B. Accuracy

Table IV summarizes the quantitative evaluation results using the KITTI benchmark metric.

The largest gain in accuracy is made by decreasing the grid cell size as for Net 5. However, also the box encoding has a large impact on the accuracy. While in Net 6 angles can not be recovered robustly the angle encoding in B 1 yields better results. Unfortunately, the network training for box encoding B3 did not converge at all. This might be due to an issue during data augmentation when boxes (and grid maps) are rotated. Also, the input features have an impact on the detection accuracy. It seems that normalization via the decay rate model yields better results that using the number of detections and observations directly. This is advantageous as the amount of grid map layers can be decreased this way. Ground surface removal has a minor impact on the detection of cars and other large objects but leads to a reduced accuracy in the detection of cyclists and pedestrians. We believe that this is due to detections close to the ground surface that are removed.

Our test results (submitted as *TopNet* variants) are similar to the validation results, yielding state-of-the-art benchmark results. This shows that no overfitting on the validation data occurred, likely due to our data augmentation strategies.

Figure 3 depicts two scenarios for qualitative comparison of three network configurations.

C. Inference Time

We evaluated the processing times on a 2.5 GHz six core Intel Xeon E5-2640 CPU with 15 MB cache and an NVIDIA GeForce GTX 1080 Ti GPU with 11 GB graphics memory. In comparison to the other networks Net 5 has the highest inference time. This is due to the large grid size as we keep the grid map extent of $80\,m \times 80\,m$ fixed across different evaluations. Net 2 has a slightly short inference time due to the R-FCN meta architecture. Using the InceptionV2 feature extractor in Net 3 also decreases the inference time compared to using a Resnet101. A different number of grid map layers or different box encodings have no significant impact on the inference time.

VI. CONCLUSION

We presented our approach to object detection and classification based on multi-layer grid maps using deep convolutional networks.

By specifically adapting preprocessing, input features, data augmentation, object encodings and proposal generation to the grid map domain we show that our networks achieve state of the art benchmark results by only using multi-layer grid maps from range sensor data. We identify the input feature selection together with the resolution as an important factor for network accuracy and training / inference time.

As a next step we aim to develop a framework for semi-supervised learning of object detectors, hopefully increasing generalization and thus overall robustness. Finally, we want to develop a tracking framework based on grid maps by coupling detections with predictions in an end-to-end learnable framework.

REFERENCES

[1] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[2] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter, and K. Dietmayer, "Fusion of Laser and Radar Sensor Data with a Sequential Monte Carlo Bayesian Occupancy Filter," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1074–1081.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," may 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," dec 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

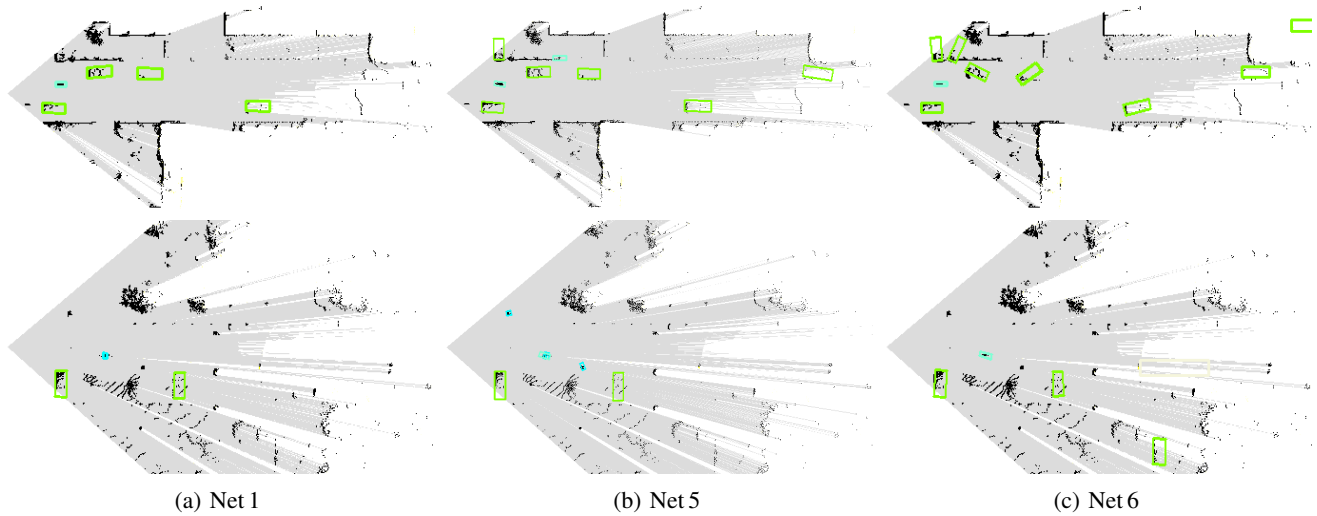|		
(a) Net 1	(b) Net 5	(c) Net 6

Figure 3: Qualitative results for three different networks on two scenarios. Cars are depicted green, cyclists aquamarin and pedestrians cyan. Compared to Net 1, Net 5 is able to detect pedestrians and distant cars. Due to the box encoding in Net 6 the rotation regression is less robust compared to Net 1 and Net 5. Best viewed digitally with zoom.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," feb 2016.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," apr 2017.

[9] S. Hinz, "Detection and Counting of Cars in Aerial Images," in *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, vol. 3, 2003, pp. III–997–1000 vol.2.

[10] T. T. Nguyen, H. Grabner, H. Bischof, and B. Gruber, "On-line Boosting for Car Detection from Aerial Images," in *2007 IEEE International Conference on Research, Innovation and Vision for the Future*, 2007, pp. 87–95.

[11] S. Kluckner, G. Pacher, H. Grabner, H. Bischof, and J. Bauer, "A 3D Teacher for Car Detection in Aerial Images," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[12] S. Wirges, F. Hartenbach, and C. Stiller, "Evidential Occupancy Grid Map Augmentation using Deep Learning," *ArXiv e-prints*, 2018.

[13] T. Zhao and R. Nevatia, "Car Detection in Low Resolution Aerial Images," *Image and Vision Computing*, vol. 21, no. 8, pp. 693–703, 2003.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pp. 3354–3361, 2012.

[15] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," *CVPR*, pp. 1907–1915, 2017.

[16] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," *ArXiv e-prints*, 2017.

[17] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based Recognition of 3d Point Clouds in Urban Environments," *IEEE 12th International Conference on Computer Vision, 2009*, no. Iccv, pp. 2154–2161, 2009.

[18] P. Babahajiani, L. Fan, and M. Gabbouj, "Object Recognition in 3D Point Cloud of Urban Street Scene," C. V. Jawahar and S. Shan, Eds. Cham: Springer International Publishing, 2015, pp. 177–190.

[19] S. Hoermann, P. Henzler, M. Bach, and K. Dietmayer, "Object Detection on Dynamic Occupancy Grid Maps Using Deep Learning and Automatic Label Generation," *ArXiv e-prints*, 2018.

[20] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection," 2017.

[21] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org.

[22] A. Schaefer, L. Luft, and W. Burgard, "An Analytical Lidar Sensor Model Based on Ray Path Information," *IEEE International Conference on Robotics and Automation*, vol. 2, no. 3, pp. 1405–1412, 2017.

[23] T. L. Kay and J. T. Kajiya, "Ray Tracing Complex Scenes," in *ACM SIGGRAPH Computer Graphics*, vol. 20, no. 4, 1986, pp. 269–278.

[24] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," feb 2015.