Deep End-to-end 3D Person Detection from Camera and Lidar

Markus Roth[†,1,2], Dominik Jargot[2] and Dariu M. Gavrila[2]

***Abstract*— We present a method for 3D person detection from camera images and lidar point clouds in automotive scenes. The method comprises a deep neural network which estimates the 3D location and extent of persons present in the scene. 3D anchor proposals are refined in two stages: a region proposal network and a subsequent detection network.**

For both input modalities high-level feature representations are learned from raw sensor data instead of being manually designed. To that end, we use Voxel Feature Encoders [1] to obtain point cloud features instead of widely used projection-based point cloud representations, thus allowing the network to learn to predict the location and extent of persons in an end-to-end manner.

Experiments on the validation set of the KITTI 3D object detection benchmark [2] show that the proposed method outperforms state-of-the-art methods with an average precision (AP) of 47.06% on moderate difficulty.

I. Introduction

Person detection plays an important role in environment perception of urban traffic scenes. *3D* person detection aims at finding the 3D location and extent of persons from one or multiple sensors. The knowledge about surrounding persons is essential to active safety systems, which aim at reducing traffic accidents, e.g., by introducing a braking manoeuvre in case of a potential collision. In self-driving cars, 3D person detection results are crucial for applications such as the reliable and safe planning of its trajectory towards the destination. Furthermore, it offers a base building block for *understanding* humans around the ego-vehicle, by means of head pose estimation, gesture recognition and intention estimation.

Detecting persons in urban traffic scenes faces many challenges, such as differing appearance and pose. Also, weather and illumination, as well as occlusions, make 3D person detection particularly challenging.

In intelligent transportation systems, different sensors are commonly used for person detection, such as cameras, radar and lidar sensors, each coming with their individual advantages and drawbacks. While a monocular camera-based system offers a dense projection of the light in the field if view, it lacks distance information. Lidar sensors offer a sparse scan of the environment with precise distance information, even during nighttime. However, even modern lidar sensors offer only 128 vertical layers.

In the last two decades, person detection performance has significantly improved due to machine learning methods and the rise of large and representative datasets to optimize and evaluate the methods on [2,3]. In the past years, deep learning methods have shown to be very accurate in the task of *2D* person detection in camera images, which in contrast to *3D* person detection, does not estimate the distance of a person to the camera sensor. 3D person detection remains more challenging, compared by the detection performance on standard benchmarks, such as the KITTI 3D Object Detection Evaluation [2]. Methods performing solely on 3D points clouds, or additionally taking camera images as an input still perform mediocre with an average precision (AP) of 46.6% on the moderate subset of the KITTI benchmark. For multi-modal methods, the problem of sensor fusion has to be solved. Some methods rely on transforming the point cloud to an image-alike structure, such as bird eye view (BEV), to employ methods known from image recognition.

Hand-crafted preprocessing of lidar point clouds raises questions such as “are there representations which perform better?”. This question is less likely to be raised for end-to-

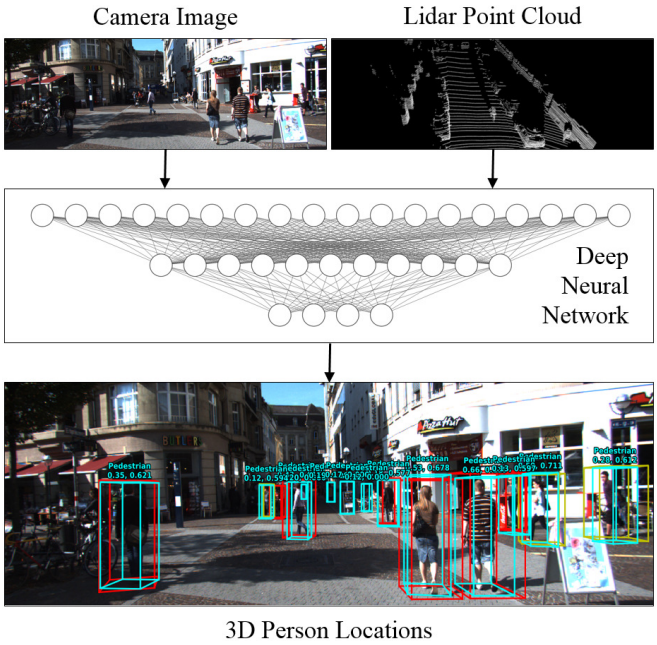

Fig. 1: The proposed method estimates 3D person bounding boxes from camera images and 3D lidar point clouds. The deep end-to-end model learns low-level features for both modalities, fuses their higher-level representations and predicts the 3D location of the persons in the scenes. Predictions are depicted by cyan bounding boxes which are projected in the camera image. Ground truth bounding boxes are shown in red. Evaluations are performed on the KITTI dataset [2].

[†] markus.r.roth@daimler.com
[1] Environment Perception, Daimler AG, Stuttgart, Germany
[2] Intelligent Vehicles, Technical University Delft, The Netherlands

end learning, where sensor input is kept as raw as possible to have representations being *learned* in contrast to being *designed*.

In this paper, we present a deep learning based method for 3D person detection, which performs end-to-end learning on sensor data from camera and lidar. High level representations from image and point cloud are learned from the raw sensor data. For the image input, a VGG-like convolutional neural network extracts a high-level feature representation. For the point cloud input, a Voxel Feature Encoder (VFE) is employed for abstract features extraction. Features from both modalities are fused to serve as an input for a regression model which estimates the 3D positions of persons.

See Figures 1 and 2 for an overview of the proposed system.

II. Related Work

We first review deep learning based object detection methods, followed by methods for 3D object detection applied to detection of traffic participants based on camera-only, lidar-only and both modalities combined.

A. Deep Learning based Object Detection

Deep neural networks, specifically convolutional neural networks (CNNs) have been shown to be very accurate in many image recognition tasks such as image classification [4], object detection and especially person detection [3]. A large number of artificial neurons stacked in several layers create a neural network. It transforms the input datum into an output datum which represents the task to solve. Within each layer, a more high-level representation of the input is encoded.

For the task of object detection, there are two different approaches, namely *two-stage* and *single-stage* object detectors.

1) Two-stage Object Detection: Two-stage object detection architectures consist of a region proposal stage and a proposal classification stage. The region proposal stage generates proposals which are to be classified by the proposal classification stage. Using a fast region proposal stage allows to keep inference time low, while still maintaining a high accuracy. Popular methods adopting this approach are Region-based Convolutional Neural Networks (R-CNN) [5], Fast R-CNN [6], Faster R-CNN [7] and Region-based Fully Convolutional Network (RFCN) [8].

2) Single-stage Object Detection: Single-stage object detection architectures perform the detection task in one forward pass through the network. You Only Look Once (YOLO) [9] and Single-shot multibox detector (SSD) [10] are prominent representatives of single-stage object detectors.

YOLO and its improved derivatives [11,12] divide an input image into a grid. Each grid cell predicts a fixed number of bounding boxes with an associated confidence score. Each bounding box is classified. The resulting detections are obtained by a non-maximum suppression (NMS).

SSD uses a base convolutional neural network (CNN) to extract feature maps at different layers. Each layer produces detection proposals based on default bounding boxes associated to each feature map. This allows for specialized classification of objects in various sizes.

B. 3D Person Detection in the Automotive Context

3D person detection in the automotive context can be performed on measurements acquired by different on-board sensors such as cameras, radar and lidar. Methods can perform either on a single modality or a combination of sensor inputs. We focus on camera-only methods, lidar-only methods and methods working on both modalities.

1) Camera-based Methods: [13] estimates the 3D pose of objects from a single image. A state-of-the-art 2D detector is used to obtain 2D bounding boxes of the objects. A CNN estimates the 3D pose of the object while considering projective geometry constraints. 3DOP [14] generates 3D object proposals from stereo-based depth information. An energy function is formulated to exploit different features such as prior object size, free-space, and point densities inside the bounding box. The 3D object proposals are then scored by a CNN. In contrast, Mono3D [15] creates 3D object proposals from monocular images by exploiting constraints such as objects residing on the ground plane. Proposals are scored by semantic information, context, as well as shape features and location priors. [16] introduces Deep MANTA, a CNN which estimates 2D bounding boxes and vehicle part locations, along with visibility and a 3D CAD template. The pose in terms of location and orientation is recovered by using a 2D/3D point mapping.

2) Lidar-based Methods: In contrast to images, point clouds are inherently unordered and have a varying size. To overcome the this issue, different representations have emerged: bird eye view (BEV) [17,18], sensor-view [19], mixed 2.5D BEV images [20], and voxel grids [1]. These representations allow for transplanting image-based methods to point clouds, specifically CNNs [18,19].

In [19], an image-like 2D point map representation is used on which a CNN is employed for 3D object detection. Complex-YOLO [18] expands the YOLOv2 CNN [11] and applies it on a BEV representation of the point cloud to detect 3D objects. These methods rely on *designing* a good representation of point clouds.

In contrast, there there are methods which *learn* features from point clouds without a strongly enforced intermediate representation [1,20]–[23]. PointNet [21] presents a permutation-invariant deep neural network which learns global features from unordered point clouds. The method is applied to 3D part segmentation and point-wise semantic segmentation. PointPillars [20] uses the features from PointNet in order to learn a feature representation on vertical columns (pillars). This allows for an image-like representation on which an SSD-based detection network is applied for 3D object detection.

Voxelnet [1] avoids using hand-crafted features by partitioning the input space into equally-sized voxels. The

group of points within each voxel is transformed into a unified feature representation through voxel feature encoding (VFE) layers. A VFE layer combines point-wise features with a locally aggregated feature. Stacking VFE layers allows for learning higher level features. The resulting high-dimensional volumetric representation is used in a region proposal network framework to estimate the 3D location of objects.

3) Camera- and Lidar-based Methods: There are multi-modal fusion methods which combine camera images and lidar point clouds. At the cost of needing a well-synchronized and calibrated sensor setup, benefits from each modality can be leveraged. E.g. small and far-away objects can be visible in the camera image while they may not have a lidar measurement.

Frustum PointNets [24] uses a state-of-the-art 2D object detector on camera images to obtain points which reside in the object's frustum. Then, 3D object instance segmentation and bounding box regression is performed on point features extracted with the method of PointNet [1].

Multi-View 3D (MV3D) [17] and Aggregate View Object Detection (AVOD) [25] are both sensor fusion methods, i.e. they take input from both camera images and lidar point clouds, extract features from each modality, fuse the features and consequently perform 3D bounding box regression.

MV3D [17] represents point clouds in a front view and a BEV image. Along with the camera image, convolutional layers are applied for high-level feature representation. A region proposal network creates view-specific feature crops from the BEV. The per-modality region-based features are fused either *early*, *late*, or in a *deep* fusion scheme. 3D bounding box regression is performed to obtain the object's 3D position.

AVOD [25] provides a similar architecture as MV3D. However, it fuses features from the individual sensors earlier in the region proposal network in order to capture smaller objects. Furthermore, AVOD uses the camera image and BEV input only, while still achieving a higher accuracy on the KITTI 3D detection benchmark.

Both MV3D and AVOD present multi-modal architectures for 3D object detection from camera and lidar. However, the methods rely on a dense image-like feature representation of the inherently sparse point cloud. To close this gap, we present an architecture which employs learned features from raw point clouds in and end-to-end framework for 3D person detection.

III. Proposed Approach

We present an end-to-end method for 3D person detection based on camera images and lidar point clouds [26]. Our proposed approach builds upon the architecture of Aggregate View Object Detection (AVOD) [25]. As in AVOD, feature maps from both a camera and lidar modality are extracted. A region proposal network (RPN) generates 3D region proposals based on cut-outs of the feature maps of 3D anchors. The top region proposals are refined by a second stage detection network which estimates the 3D location and extent of the persons present in the scene.

In contrast to AVOD, which relies on hand-crafted bird-eye-view (BEV) for the lidar input, the presented approach learns point cloud features by applying Voxel Feature Encoding (VFE) layers followed by 3D convolutional layers for high level feature extraction as introduced in VoxelNet [1]. The proposed architecture is depicted in Figure 2.

A. Contributions

Our main contributions are three-fold: (a) we introduce a novel end-to-end deep learning based method for 3D person detection using camera images and lidar point clouds, (b) the proposed method does not rely on hand-crafted features, and (c) outperforms the existing state-of-the-art on the validation dataset of the KITTI 3D object detection benchmark [2].

B. Input Preprocessing and Feature Extraction

Both camera images and lidar point clouds are preprocessed to allow for subsequent feature extraction.

1) Image Preprocessing: The RGB camera images are normalized by subtracting the mean RGB value of the training dataset. For image feature extraction, we use the VGG16 architecture [27] with the same modifications as in [25], i.e. half the number of filters in each convolutional layer, no fifth convolutional stage and no max-pooling layer at the end of the fourth stage. The resulting 256 feature maps are 8 times smaller along each dimension. To attain higher resolution feature maps, 4 times bilinear upsampling is applied.

2) Point Cloud Preprocessing: The lidar point cloud is cropped to reside in the volume $\Delta X = [-40\text{m}, 40\text{m}], \Delta Y = [-1\text{m}, 3\text{m}], \Delta Z = [0\text{m}, 70\text{m}]$ in the camera frame. The volume is partitioned into equally sized voxels of size $(s_x, s_y, s_z) = (0.2\text{m}, 0.4\text{m}, 0.2\text{m})$. The voxelized input is processed by a Feature Learning Network, as proposed in [1]. It consists of grouping, random sampling and stacked voxel feature encoding (VFE) layers. Each point $p_i = (x_i, y_i, z_i)$ in each voxel v forms an input vector $(x_i, y_i, z_i, r_i, \tilde{x}_v, \tilde{z}_v, \tilde{z}_v)$ with point reflectance r_i and voxel mean coordinates $(\tilde{x}_v, \tilde{z}_v, \tilde{z}_v)$. Each VFE layer learns a locally aggregated feature by a fully connected layer on the input, followed by max-pooling and concatenation [1]. We randomly sample $T = 45$ points per voxel and stack 2 VFE layers. The first VFE layer yields a 32 dimensional feature vector per voxel. The second VFE layers yields a 128 dimensional feature vector per voxel.

Three 3D convolutional layers conv3D(c, k, s, p) with output channels c, kernel size k, stride $s = (s_x, s_y, s_z)$ and padding $p = (p_x, p_y, p_z)$ aggregate voxel-wise features to obtain an expanding receptive field to capture more context information [1]. The parameters of the convolutional layers are:

$$\text{conv3D}_1\,(64, (1, 2, 1), (1, 1, 1))$$
$$\text{conv3D}_2\,(64, (1, 1, 1), (1, 0, 1))$$
$$\text{conv3D}_3\,(64, (1, 2, 1), (1, 1, 1))$$

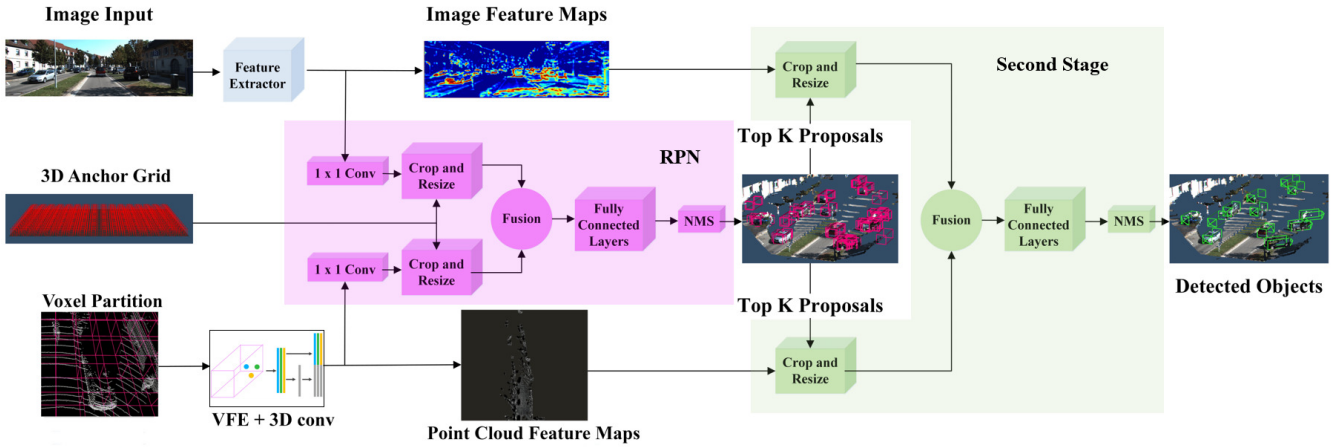

Fig. 2: The architecture of the proposed method [26]. It is inspired by AVOD [25]. Image features are extracted by an adapted VGG16 network. Point cloud features are extracted from voxel partitions by applying Voxel Feature Encoding (VFE) layers and 3D convolutions. In a Region Proposal Network (RPN), 1×1 convolution is applied to the feature maps to reduce their size. Anchors from a 3D anchor are projected into the feature maps to crop proposals. After resizing to a common size, the feature crops from both modalities are fused and a the location of the objects is refined by a fully connected neural network. In the second stage, the best proposals from the RPN are cropped from the full feature maps and fused. Object detection layers are implemented by fully connected layers operating on the fused crops. This allows for an end-to-end network which estimates the 3D locations of persons from camera and lidar sensor data. Image adapted from [1,25].

Reshaping is performed such that neighboring voxels along y dimension are flattened, thus resulting in a voxel feature map with 128 channels per voxel.

3) Anchor Generation: As in [25], anchors are spawned in a 3D dense grid in the voxel volume, using an interval of 0.5m along x and z direction and y coordinate to reside on the ground plane. The extents of the anchors are based on size clusters obtained for each class on the training set. Anchors which are outside the camera view or not supported by any point are discarded.

C. Region Proposal Network

The region proposal network (RPN) projects anchors to the feature maps of each modality, crops the respective feature residing in the anchor projections, resizes them and fuses them. Subsequently, fully connected layers refine the location of anchor boxes towards ground truth location to form the region proposals. The RPN is adopted from [25], but in contrast our approach crops from the learned voxel feature map instead of the bird eye view feature map. For the RPN, the feature maps are reduced in dimensionality by performing a 1×1 convolution [25] which can be seen as a learned weighting of all feature maps along the y dimension.

We use crops of size 3×3 from the feature maps of each modality and retain the 1024 best proposals after non-maximum suppression. The crops are fused using the mean operation. There are two fully connected layers with 2048 neurons each.

Proposals to optimize are selected by having an intersection over union (IoU) of > 0.8 with the projected ground truth box. We use a Smooth L1 loss for localization regression task and a cross-entropy loss for the classification task (person vs. background).

D. Detection Network

The second stage detection network is also based on [25], i.e. we crop the region proposals from the feature maps of both modalities, fuse them and regress location, extent and class by fully connected layers.

We use crops of size 7×7 from the feature maps of both modalities, and concatenate them. There are three fully connected layers with 2048 neurons each.

The location and extent of the detected persons are retained after a non-maximum suppression.

E. Fusion Schemes

Both the RPN and the detection network fuse resized feature map crops from both modalities. MV3D [17] proposes three different fusion schemes, namely *early*, *late*, and *deep* fusion. Combinations of individual input can be *concatenation* or *element-wise mean*.

The fusion schemes differ in which order feature transformations (e.g. convolutions) are applied compared to feature combinations. *Early fusion*: combine individual inputs, then transform. *Late fusion*: transform inputs, then combine. *Deep fusion*: combine inputs, then transform individually, and repeat. In deep fusion, the transformations of each repetition learn different parameters.

F. Training

We train the proposed end-to-end network by using the ADAM optimizer [28]. One scene per training iteration is used, yielding 1024 proposals for training the network. The RPN and detection network are trained jointly. We trained the network on the *train* split starting from a random initialization. The learning rate is set to 0.0001.

TABLE I: Statistics of the KITTI dataset used for evaluation. Each scene represents a synchronized snapshot of the environment at a point in time. The number of 3D annotations in the camera's field of view is listed.

Dataset	# Scenes	# 3D Annotations		
		Cars	Cyclists	Pedestrians
KITTI train	3712	14357	734	2207
KITTI val	3769	14385	893	2280
KITTI test	7518	-	-	-

IV. Experimental Evaluation

We evaluate the proposed method on the pedestrian class of the KITTI 3D Object Detection Evaluation 2017 (KITTI) [2] using average precision (AP), which follows the standard evaluation protocol of the KITTI benchmark.

A. Dataset

The KITTI dataset [2] captures 15k urban traffic scenes by camera images and lidar-based point clouds. Traffic participants such as cars, cyclists and pedestrians are annotated by 3D bounding boxes. Three difficulty levels are defined (easy, moderate, hard) based on object size in the camera image, occlusion state and truncation ratio.

We use the annotated scenes and divide them into a training split (*train*) and validation split (*val*), as in [14], which ensures that images from the splits are from disjoint sequences. See Table I for an overview of the KITTI dataset and the provided annotations.

B. Data Augmentation

As the KITTI *train* split offers around 4k scenes with around 2.2k pedestrians, we use augmentations to increase diversity in the training set. We flip the image along the vertical axis and the corresponding point cloud along the yz plane. The principal point of the camera matrix is adapted accordingly to ensure valid projections into the flipped camera image.

C. Evaluation Metrics

To evaluate the performance of the 3D object detection task, we use average precision (AP), as defined in [2,29], i.e.

$$\frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} \max_{\text{recall}(c) \geq r} \text{precision}(c)$$

with $\text{recall}(c) = \frac{\text{tp}(c)}{\text{tp}(c)+\text{fn}(c)}$, and $\text{precision}(c) = \frac{\text{tp}(c)}{\text{tp}(c)+\text{fp}(c)}$, both for an objectness confidence threshold c. $\text{tp}(c)$ and $\text{fn}(c)$ denote the number of true positives and false negatives, respectively.

A 3D prediction bounding box is considered to correspond to a 3D ground truth annotation bounding box, if the intersection over union (IoU) in xz coordinates is above 0.7. This follows the evaluation protocol of the KITTI 3D Object Detection Evaluation 2017 [2].

TABLE II: Selected experiments of hyper parameters and their performance on the *val* split of KITTI.[1]

Exp.	RPN Stage		Detection Network			Ped. AP (%)		
	Comb.	Crop	Fus.	Comb.	Crop	Easy	Mod.	Hard
#1	mean	3×7	early	mean	7×7	45.85	40.79	35.92
#2	mean	3×3	deep	mean	7×7	44.18	37.11	30.36
#3	mean	3×3	late	mean	7×7	49.56	43.68	38.36
#4	mean	5×5	early	mean	9×9	50.00	44.47	38.70
#5	concat	3×3	early	mean	7×7	51.91	46.38	40.86
#6	mean	3×3	early	concat	7×7	53.29	46.23	40.28
#7	mean	3×3	deep	concat	7×7	**53.47**	**47.06**	**41.49**

D. Experimental Results

We use the *train* split to train our model using different hyper parameters and evaluate on the *val* split to evaluate the performance using AP. The hyper parameters under test were fusion schemes and combination methods for both the RPN and the second stage detection network, as introduced in Section III-E. Additionally, we varied the feature crop size.

1) Quantitative Analysis: All models were trained starting from random initialization and continuously evaluated on the *val* split every 1000 training iterations, stopping at maximum 120k iterations. For all comparisons, the best-performing model over all training iterations was chosen.

Table II shows selected experiments which we analyzed and their respective performance on the *val* split. Using the late fusion scheme in the detection network yields a higher performance than deep or early fusion when combining the individual features via an element-wise mean (experiments #1 - #3). Increasing the crop sizes increases AP when keeping element-wise mean (experiment #1 vs. #4). The highest performing experiments are using concatenation feature combination in the detection network (experiments #6 & #7). Among those experiments the deep fusion scheme performs slightly better than early fusion scheme. We therefore chose the model from experiment #7 for further experiments.

Table III shows quantitative performance of the proposed model compared to selected state-of-the art methods on the KITTI 3D Detection Benchmark. We chose F-PointNet [24] and PointPillars [20] as they are among the best performing methods on the KITTI benchmark[2]. VoxelNet [1] and AVOD [25] are of special interest for comparison, as the former uses the same feature extraction for point cloud features and the latter introduced the architecture our proposed method is based on.

As can be seen, the proposed method outperforms the baselines on all three difficulties.

2) Qualitative Analysis: We present 3D detection examples in Figure 3. The detected 3D bounding boxes are projected into the camera images.

Figures 3a to 3d represent examples with good detection performance. The proposed model detects persons in

[1]As typical for high dimensional models, alterations in the hyper parameter space have a highly non-linear impact on the model performance. We therefore only show a subset of the conducted experiments with good performance.

[2]`http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d`, retrieved 2019-04-09

TABLE III: Comparison of the proposed method vs. vs state-of-the-art. Results on the KITTI 3D detection benchmark. Test split is used, unless otherwise given. Numbers are given in % average precision (AP) for IoU > 0.7.

Method	Modality	Pedestrian		
		Easy	Mod.	Hard
F-PointNet [24]	lidar	51.21	44.89	40.23
PointPillars [20]	lidar	52.08	43.53	41.49
VoxelNet [1]	lidar	39.48	33.69	31.51
AVOD [25]	lidar & image	38.28	31.51	26.98
Proposed (*val*)	lidar & image	**53.47**	**47.06**	**41.49**

crowded scenes (Figure 3a), as well as far away objects (Figure 3b). The model is robust against occlusions and truncations on the image border (Figures 3c and 3d).

Limitations of the system are presented in Figures 3e to 3h. While the system detects some highly occluded persons, it misses others and creates false positives (Figure 3e). Figure 3f shows a traffic structure which is falsely detected as a person. Figures 3g and 3h show confusions with cyclists.

V. Conclusion

We presented a novel deep end-to-end method for 3D person detection from camera images and lidar point clouds. The method does not rely on hand crafted features. Instead, it learns high-level features from both camera images and lidar point clouds. Point cloud features are extracted using voxel feature encoders.

Experiments on the KITTI 3D object detection benchmark show that the presented method outperforms the existing state-of-the-art with an average precision of 47.06% on moderate difficulty.

The method relies on anchor proposals which reside on the ground plane. It is therefore dependent on a robust ground plane estimation algorithm. We see possible improvements in finding 3D anchor proposals which are independent of the ground plane. Furthermore, the method may benefit from being trained on a larger-scale dataset.

References

[1] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," arXiv:1711.06396, 11 2017.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2012, pp. 3354–3361.

[3] M. Braun, S. Krebs, F. Flohr, and D. Gavrila, "EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1844–1861, 5 2019.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2014, pp. 580–587.

[6] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 12 2015, pp. 1440–1448.

[7] 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, 6 2017.

[8] J. Dai, L. Yi, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2016, pp. 779–788.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 6517–6525.

[12] ——, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 4 2018.

[13] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D Bounding Box Estimation Using Deep Learning and Geometry," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 5632–5640.

[14] X. Chen, "3D Object Proposals for Accurate Object Class Detection," *Advances in Neural Information Processing Systems (NIPS)*, pp. 424–432, 2015.

[15] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D Object Detection for Autonomous Driving," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2147–2156, 2016.

[16] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep MANTA: A Coarse-to-Fine Many-Task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 1827–1836.

[17] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D Object Detection Network for Autonomous Driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 6526–6534.

[18] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-YOLO: Real-time 3D Object Detection on Point Clouds," arXiv:1803.06199, 3 2018.

[19] B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," in *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016.

[20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," arXiv:1812.05784, 12 2018.

[21] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 77–85.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," arXiv:1706.02413, 6 2017.

[23] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, vol. 18, no. 10, p. 3337, 10 2018.

[24] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," arXiv:1711.08488, 11 2017.

[25] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," arXiv:1712.02294, 2017.

[26] D. Jargot, "Deep End-to-end Network for 3D Object Detection in the Context of Autonomous Driving," *MS Thesis TU Delft*, 2019.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980, 12 2014.

[29] M. Everingham, L. Van˜Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3: Qualitative results for the proposed model on the KITTI validation set. The model can estimate the 3D location of people with different orientations and poses. (a) - (d): examples with good performance on crowded scenes, far away objects, occluded and truncated objects. (e) - (h): failure cases with missed detections, false positives and confusions with cyclists. Ground truth: red. Predictions: cyan. Numbers on top of boxes: detector objectness, IoU.