IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 3, 2022, accepted May 29, 2022, date of publication June 8, 2022, date of current version June 14, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3181137*

# Significance of Image Features in Camera-LiDAR Based Object Detection

MIHÁLY CSONTHÓ, ANDRÁS RÖVID, AND ZSOLT SZALAY
Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, 1111 Budapest, Hungary

Corresponding author: Mihály Csonthó (csontho.mihaly@kjk.bme.hu)

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

**ABSTRACT** In autonomous cars accurate and reliable detection of objects in the proximity of the vehicle is necessary in order to perform further safety critical actions which depend upon it. Many detectors have been developed in the last few years, but there is still demand for more reliable and more robust detectors. Some detectors rely on a single sensor, while some others are based upon fusion of data from multiple sources. The main aim of this paper is to show how image features can contribute to performance improvement of detectors which rely on pointcloud data only. In addition it will be shown, how lidar reflectance data can be substituted by low level image features without degrading the performance of detectors. Three different approaches are proposed to fuse image features with point cloud data. The extended networks are compared with the original network and tested on a well-known dataset and on our own data, as well. This might be important when the same pretrained model is to be used on data generated by a lidar using different reflectance encoding schemes and when due to the lack of training data retraining is not possible. Different augmentation techniques have been proposed and tested on the KITTI dataset as well as on data acquired by a different lidar sensor. The networks augmented with image features achieved a recall increase of a few percent for occluded objects.

**INDEX TERMS** Autonomous driving, environment perception, image features, neural networks, object detection, sensor fusion.

## I. INTRODUCTION

In the field of autonomous driving and intelligent infrastructures the environment perception stands for a safety critical task where different type of static and dynamic objects must reliably and robustly be detected and localized under various circumstances, such as different weather conditions, limited sensing resolution of applied sensors, partial occlusions, etc.

Efficient sensing under different weather conditions might easily be handled by utilizing different type of sensors jointly (lidar, RADAR, Camera, thermal vision). Most often camera and Lidar sensors are used in sensor fusion algorithms [1], [2]. For various applications, camera and radar pairing is also common [3], [4], while there are also cases where radar and lidar data are considered for fusion. [5].

Lidar sensors are not affected by day and night lighting conditions, they can also reliably operate under various limited visibility conditions. Radar sensors are also unaffected by light and weather conditions such as fog and dust, they can sense longer distances than Lidars, however their resolution compared to Lidars is considerably lower. Cameras perform poorly under limited visibility conditions, although thermal imaging cameras can compensate for such limitations. [6]

Individual application of certain sensors might strongly be limited by their low spatial resolution as in case of lidars for instance (depending on the displacement, number of channels and the field of view different sparse patterns can be observed on the generated pointcloud). Even the most advanced lidars are not able to capture objects being at longer distances (> 150 m) with good enough resolution which makes the detection task in such cases even more difficult. At distances more than 150 the number of rays crossing the body of an average sized vehicle is to low for its reliable detection (even in case of lidars having the highest available resolution).

There are numerous cases when long range detection of vehicles is required in order to perform the given task efficiently, such as for instance prediction of potentially

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

dangerous traffic situations or scenarios in order to avoid accidents and increase road safety [7]; digital twin generation of longer road sections, where the range of detectability influences also the minimal number of sensors to be deployed in order to cover a given road section. This factor plays significant role first of all due to cost, energy consumption and maintenance related reasons in future intelligent infrastructure and road networks [8].

Multi-modal approaches are a promising alternative to handle (to some extent) problems caused by the sparsity of lidar pointclouds. For example by combining information from camera images (which obviously have much larger resolution than lidars) with pointcloud data (which on the other hand have good depth resolution) the detection performance as well as the reliability might be improved (compared to camera only or lidar only approaches).

The main contribution of this paper is represented by the proposed pointcloud augmentation techniques incorporated into a selected baseline lidar based detector model and by the evaluation and analysis of the impact of certain image features on 3D object detection performance compared to the lidar only detection case where the training of neural networks as well as the inference is solely performed on pointclouds. It is also examined how certain types of image features under different conditions (various scenarios including partially occluded, close and distant vehicles, data acquired by various lidar types) contribute to the performance improvement of lidar only based solutions by transforming the pointcloud into a ''clever'' pointcloud by applying the proposed augmentation techniques.

The paper is organized as follows:

- in Section II the related work including the brief overview of the state of the art solutions of sensor fusion is described;
- Section III summarises the problem addressed by the point cloud augmentation algorithms presented;
- Section IV presents the proposed point cloud augmentation algorithms;
- Section V analyses the results achieved by using augmentation networks;
- Finally Section VI reports conclusions.

## II. RELATED WORKS

Many methods appeared in the literature in the last few years (first of all machine learning based approaches) to tackle the detection problem especially in lidar pointclouds. Let us categorize the methods developed for object detection into two classes, i.e. approaches which operate on lidar pointclouds only and approaches utilizing camera images together with lidar pointclouds.

### A. LIDAR ONLY APPROACHES

Lidar only approaches are efficient for short range detection, however at longer distances the density of lidar points is significantly reduced, which makes it difficult to detect objects reliably. By utilizing lidars the vehicle or pedestrian detection task might be performed under various weather conditions efficiently. Building on the PointNet design developed by Qi *et al.* [9], VoxelNet [10] was one of the first methods to perform true end-to-end learning in this area. VoxelNet creates voxels, applies a PointNet to each voxel, followed by a 3D convolutional middle to consolidate the vertical axis, after which a 2D convolutional detection architecture is applied. While the performance of VoxelNet is robust, inference time is too slow for real-time deployment. Recently, SECOND [11] improved the inference speed of VoxelNet, but 3D convolutions remain a bottleneck. The bottleneck was solved by PointPillars [12] which is still one of the most computationally efficient architecture (according to the KITTI benchmark site [13]) designed for 3D object detection task in lidar pointclouds. In PointPillars the 3D points are organized into columns (pillars) and transformed into a sparse tensor of learnt abstract features which are then processed by further convolutional layers to get detections in form of 3D bounding boxes. A different concept for object detection in pointclouds is proposed by the authors of the so called Self-Ensembling Single-Stage object Detector (SE-SSD) where they focus on exploiting both soft and hard targets by introducing two Single-Stage object Detector (SSD) networks being in a ''student'' ''teacher'' relation. [14]. The Semantic Point Generation (SPG) method proposed in [15] aims to recover missing parts of foreground objects by generating semantic points which might be utilized by pointcloud based object detectors directly to enhance detection.

### B. CAMERA AND LIDAR BASED APPROACHES

In order to extend the range of detectability of objects and increase reliability, joint application of different sensor types is highly welcome. The authors in [1] proposed a multi-modal approach by fusing information from lidar pointclouds and semantic-rich stereo images. They bridge the resolution gap between the lidar and Camera by introducing so called virtual points. Another multi-modal approach is proposed in [2] where the the lidar points are augmented by semantic information being extracted from images in form of pixel categories resulted by semantic segmentation of the image. In the so called EPFNet [16] the authors enhance lidar points with semantic image features in a point-wise manner without any image annotations. In the work [17] the pointcloud of occluded objects is handled by learning object shape priors based on which the shape of the complete object might be estimated. Authors in [18] consider geometric consistency between detections in the image and the pointcloud, meaning that 2D bounding boxes and the projected 3D bounding boxes of detections must be consistent as well as the so called semantic consistency which is related to the category of objects. The RPN model proposed in [19] performs multi-modal fusion on high resolution feature maps in order to generate more reliable 3D object proposals for multiple object classes.

In this paper a camera-lidar fusion for object detection is proposed, which is based on augmenting the lidar points with corresponding image patterns as well as by individual pixel data. We will show how fusion strategies of this kind affect the performance of the baseline detector. The proposed fusion models enable real-time application (the frame rate of the detector is kept at 20 fps which is the frame rate of lidar sensors available today). The effectiveness of the proposed augmentation techniques is evaluated on the KITTI dataset as well as on further real world data collected by the authors. It is also shown how such augmentations may improve the performance of the selected baseline network when performing the inference on pointclouds generated by different lidar sensors.

### C. FUSION RADAR WITH CAMERAS OR LIDAR

In sensor fusion radars are also well considered sensors, first of all due to their longer range, cost efficiency and applicability even under limited visibility conditions. The so called AssociationNet [3] generates a pseudo-image from radar pins, 2D bounding boxes and the original RGB camera image which is fed into a neural network to learn high-level semantic representations. Camera-radar fusion might be applied at object level, as well [4].

Thermal imaging cameras operate well even under limited visibility conditions, they can jointly be utilized with Lidars to achieve more accurate object detection. In [20] for instance the authors use such combination of sensors while in [21] radar sensor is also included. Researchers at the University of Berlin have presented a solution where radar and Lidar detections are fused aimed for highway applications [22].

## III. PROBLEM DESCRIPTION

There have many object detectors been proposed during the last few years operating on various types of sensory data (lidar pointcloud, camera image, radar pointcloud, etc.). Here our main goal is to show how the performance of an object detector operating on pointclouds only might further be improved by low level fusion with camera images. We will also show how a network trained on data acquired by a specific sensor performs on pointcloud data acquired by comparable sensors of other vendors and what improvement in detection performance might be expected when fusion is applied. Here under fusion we mean camera-lidar fusion, i.e. fusing image pixels with pointcloud data.

Here we would like to point out the impact of sensor specific data patterns - produced by different lidar sensors - on detector performance (due to beam angles, resolution and sensitivity varying from sensor to sensor). Obviously some performance degradation of detectors might be expected due to sensor specific pointcloud patterns and reflectivity profiles representing the objects. We would like to show how camera-lidar fusion performed at lower level of abstraction may contribute to the reduction of performance degradation. Since there are many types of lidar sensors and many setups exist (each causing different pointcloud patterns to appear on surfaces of objects), collecting training data for each specific setup and sensor type individually is energy and time consuming. Instead of retraining the network on sensor or setup specific training datasets, we aim at improving its robustness by applying lower level fusion of pointclouds with image pixel data.

As baseline model we have chosen the PointPillars [12] object detector (operating on lidar pointclouds) which has remarkable performance considering its speed and precision of detections (according to the KITTI 3D object detection benchmark). Although some newer detectors managed to get higher precision but they have still much lower frame rate than PointPillars. We have trained the baseline model as well as the fusion capable network on the KITTI training dataset [23].

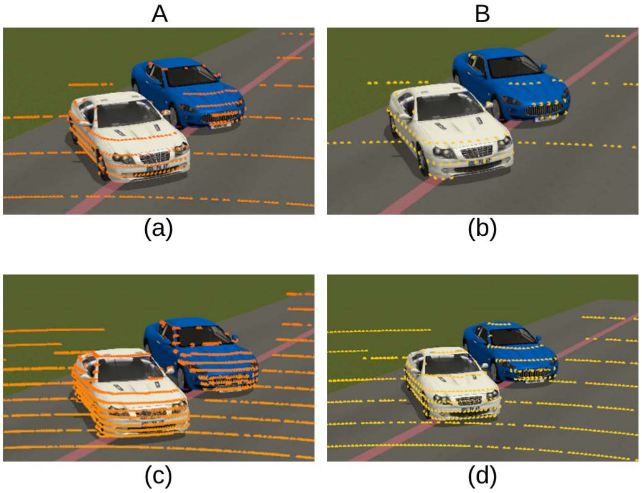
### A. DIFFERENCE IN POINTCLOUD PATTERNS

The pointcloud patterns formed on object surfaces differ from manufacturer to manufacturer of lidar sensors (by considering the same scenario and sensor placement), which may strongly influence the performance of networks trained for a specific lidar sensor but applied on data acquired by a different one. The following figure shows two pointcloud patterns corresponding to two different lidar sensors (both sensors were modeled in accordance with their specification sheets by the dSpace SensorSim sensor simulator)(see Fig. 1). Let us call these sensors as sensor-A and sensor-B. In Figs. 1a and 1b vehicles being 25m apart from the sensor origin can be followed while in figs. 1c and 1d the vehicles were set to be 15m away from the sensor origin. The height of the lidar sensor for both scenarios was set to be 1.73m (according to the test vehicle of the Karlsruhe Institute of Technology [[13], [23]]). The orientation of vehicles was 45° wrt. longitudinal axes of the lidar. The aim of this simulation is to point out the differences between pointcloud patterns. One may observe that the density of points as well as the formed pointcloud patterns differ in both cases. Another factor to be considered is the difference in reflectivity profile of lidar sensors The performance of the trained detector obviously degrades when running on data acquired by a different lidar sensor. Another important aspect here is the intensity profile of lidars, which may also differ from vendor to vendor and therefore it stands for an additional limiting factor for the usability of pretrained neural networks (trained on specific lidar data) in case of different lidars. Each manufacturer handles the reflection of the laser beam differently, from which the reflectivity value is calculated by the sensor.

In the upcoming sections we will show how the lidar reflectivity information influences the performance of the baseline model and how image pixel information may contribute to the performance improvement of detectors compared to lidar reflectivity values.

## IV. PROPOSED POINTCLOUD AUGMENTATION

In order to combine data from different sensors to generate higher level features to enhance the performance of detectors



(a) (b)

(c) (d)

**FIGURE 1. Difference between two 64 channel lidar sensors. Sensor-A corresponds to the one used in the KITTI dataset [23], while sensor-B models a sensor from different manufacturer. Images (a) and (b) show the pointclouds when the objects are located 25m from the sensor origin. Images (c) and (d) show the case when the vehicles are located 15m from the sensor origin. Significant difference can be recognized in the density of points thus in the formed patterns.**

we proposed a data driven sensor fusion approach where the fusion itself is done by neural network architectures, as well as in IMF-DNN architecture [24]. The data acquired by the lidar is combined with image features (see later in this section) which is applied during training as well as inference. The other class of fusion algorithms, where mathematical models are used to generate detections is called model-based fusion [25].

### A. THE SELECTED BASELINE MODEL

We selected the PointPillars convolutional neural network proposed by authors in [12] as the baseline model in order to apply and evaluate the impact of our proposed augmentation techniques on detection performance. The main components of the PointPillars are the so called Pillar Feature Network, the Backbone, and the Single Shot Detector (SSD) head [26]. It converts the raw pointcloud to a stacked pillar tensor and a pillar index tensor. Then a feature encoder uses the stacked pillars to learn a set of features to form a so called 2D pseudo-image serving as input for the Backbone convolutional neural network. Based on the generated features the detection head predicts 3D bounding boxes of objects present in the scene [12]. Starting from this baseline model our aim was to include image pixel information into the process of pseudo image creation in order to force the network to learn higher level features from pointcloud and image data jointly. For transforming the augmented input into a higher level feature vector (see Fig.2) a fully connected layer has been applied similarly as in [ [9], [10]]. The next section (IV-B) gives a deeper insight into the extended architecture as well as the alternatives used for image–pointcloud fusion.

### B. EXTENDED PILLAR FEATURE NET

The idea of using image features comes from the fact that when using different brands of lidars, we cannot fully align the reflectivity values. Another problem that arises is that pretrained models may be sensitive to internal sensor parameters, such as the angle or pitch of the beams. As extensions to the original baseline model, three different architectures have been proposed to increase the robustness against the influence of varying sensor parameters.

Let $\mathbf{p}_i = [u_i, v_i]^T$ stand for the pixel coordinates of the projection of a 3D point $\mathbf{P}_i = [X_i, Y_i, Z_i]^T$ from the lidar pointcloud onto the camera image plane using the pinhole camera model as follows:

$$\tilde{\mathbf{p}}_i = \mathbf{K}[\mathbf{R}|\mathbf{t}]\tilde{\mathbf{P}}_i, \tag{1}$$

where $\tilde{\mathbf{p}}_i$ and $\tilde{\mathbf{P}}_i$ stand for the homogeneous coordinates of $\mathbf{p}_i$ and $\mathbf{P}_i$, respectively, $\mathbf{K}$ denotes the camera matrix (which contains the focal length $\mathrm{f}_x$ and $\mathrm{f}_y$ expressed in terms of pixel width and height, respectively; principal point coordinates $\mathrm{x}_0$, $\mathrm{y}_0$ and the axis skew s), $\mathbf{R}$ the rotation matrix and $\mathbf{t}$ the translation vector corresponding to the transformation from the lidar frame to the camera frame. Let $I_i^L$ and $I_i^{cam}$ stand for the reflected laser beam reflectivity and the image pixel intensity of $\mathbf{P}_i$ and its projection $\mathbf{p}_i$, respectively.

Let us point out that in the baseline model [12], we augment each lidar point $\mathbf{P}_i$ in the pillar it is contained in, as follows:

$$\mathrm{P}_i^* = [X_i, Y_i, Z_i, r_i, X_i - M_x^j, Y_i - M_y^j, Z_i - M_z^j, X_i - C_x^j, Y_i - C_y^j], \tag{2}$$

where $\mathbf{M}^j = [M_x^j, M_y^j, M_z^j]$ and $\mathbf{C}^j = [C_x^j, C_y^j, C_z^j]$ denote the mean of points falling in the $j$th pillar and the center of the pillar, respectively. Considering the above original augmentation we have incorporated image pixel information into $\mathbf{P}_i^*$ as follows:

Let $\mathbf{P}_i^{**}$ denote the reduced version of the augmented point $\mathbf{P}_i^*$, where $r_i$ is not included. The following cases have been considered:

1) Each $\mathbf{P}_i^{**}$ is augmented by $v_i$ (1P1P)
2) Each $\mathbf{P}_i^{**}$ is augmented by the intensity vector formed from a $N \times N$ neighborhood of $\mathbf{p}_i$ (1P25P)
3) Each $\mathbf{P}_i^{**}$ is augmented by the normalized intensity vector formed from a $N \times N$ neighborhood of $\mathbf{p}_i$ (1P25PN)
4) Each $\mathbf{P}_i^*$ is augmented by $r_i$ and $v_i$ (1P1P)
5) Each $\mathbf{P}_i^*$ is augmented by $r_i$ and the intensity vector formed from the intensities of a $N \times N$ neighborhood of $\mathbf{p}_i$ (1P25P)
6) Each $\mathbf{P}_i^*$ is augmented by $r_i$ and the intensity vector formed from the normalized intensities of a $N \times N$ neighborhood of $\mathbf{p}_i$ (1P25PN)

During our experiments we set $N = 5$. Together with the original baseline models (with and without considering $r_i$) eight networks corresponding to the above cases were trained, evaluated and tested. Each of these networks was trained and tested on the same splits of the KITTI [13] dataset. The original training data (7481 snapshots) was split by random selection into 3212 training, 3269 validation and 1000 test samples. After evaluating the networks on the test

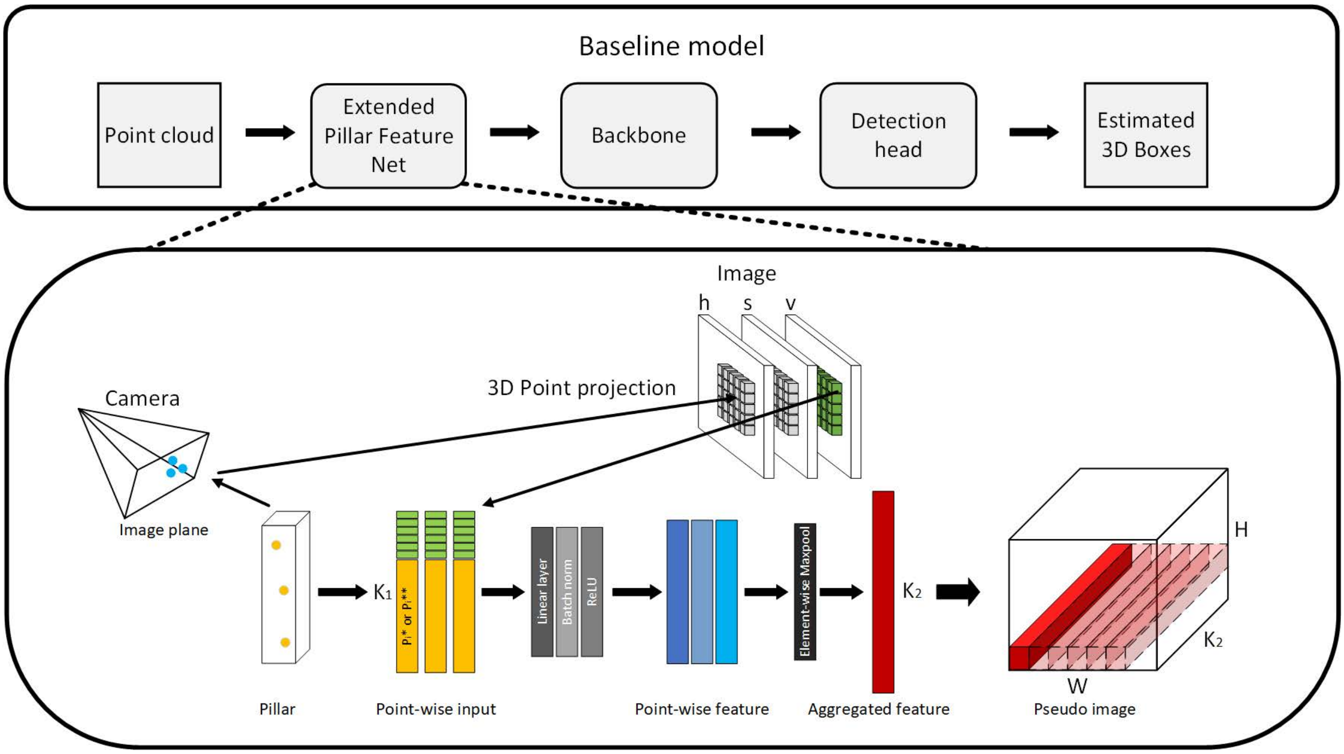


**FIGURE 2.** The PointPillars baseline model [12] and the extended pillar feature network where each 3D point is projected onto the camera image plane from where a single intensity value or the intensities in form of a vector from a $N \times N$ neighborhood of the image pixel is taken for augmentation. $K_1$ and $K_2$ stand for the number of elements in the augmented and in the aggregated feature vector respectively.

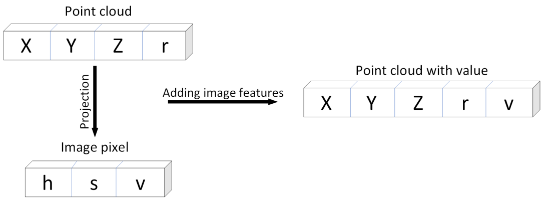


**FIGURE 3.** 1P1P architecture, where each 3D point $\mathbf{P}_i$ is augmented with the intensity of the image pixel corresponding to the projection of $\mathbf{P}_i$.

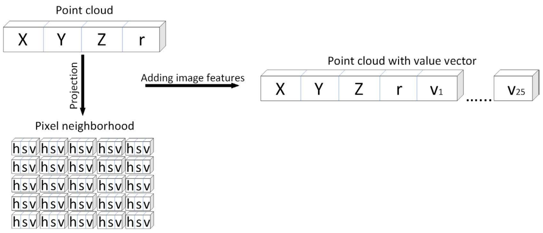


**FIGURE 4.** The 1P25P network where each 3D point $\mathbf{P}_i$ is augmented by the intensities of the 5 × 5 neighborhood of $p_i$.

set, we tested their performance on KITTI RAW [23] data as well as on data collected by us using a lidar different from the one used in KITTI. Unfortunately, there is no ground truth for raw and custom dataset, so we cannot determine the accuracy of the detections for those cases, but we can draw useful conclusions from the number of true/false detections. In the following chapters let us describe the structure of the extended feature network in detail.

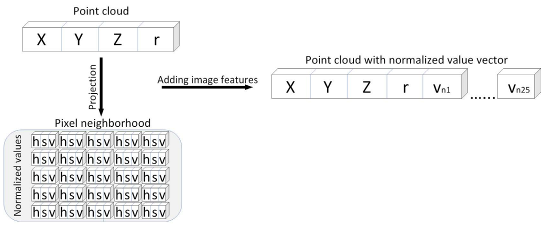


**FIGURE 5.** The 1P25PN network where each 3D point $\mathbf{p}_i$ is augmented by the normalized intensities of the 5 × 5 neighborhood of $\mathbf{p}_i$.

The modified architectures extend the PFN (Pillar Feature Network) network of the baseline model. The modified network is of size $(9 + K, 64)$, where the 9 features in the original input are augmented by $K = 1$ or $K = 25$ image features, while the output size is 64. The augmentation is performed as follows:

### 1) THE 1P1P NETWORK

First, the original network was modified by attaching to each point $\mathbf{P}_i$ in the pointcloud the intensity value (taken from the HSV color space) of the pixel corresponding to the projection of $\mathbf{P}_i$ in the camera image (see Fig.3). In order to project a 3D point onto the camera image plane the camera and the lidar must be calibrated first, i.e. the intrinsics and extrinsics must be estimated. For this purpose the calibration approaches in [[14], [27]] have been used.

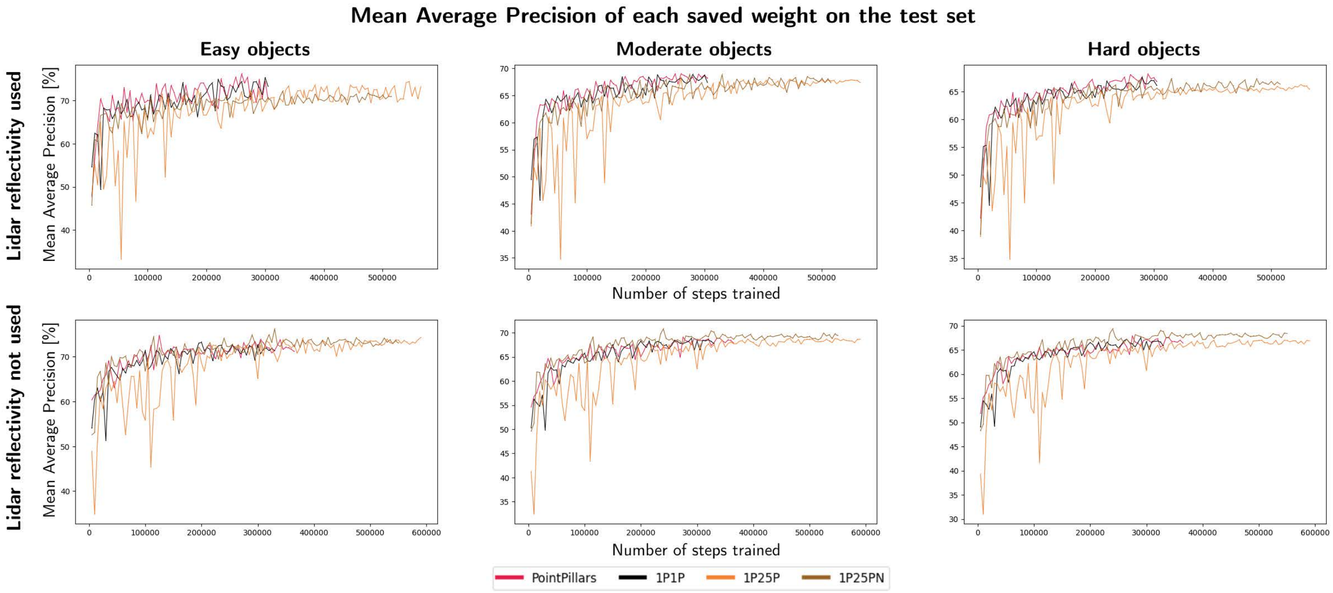


**FIGURE 6.** Mean Average Precision (mAP) values for each trained weights, saved during training. The 1st row reflects the results for the case when lidar reflectivity was considered, the 2nd row shows the results when lidar reflectivity was omitted. The 1st column shows the mAP values computed on the "Easy", the 2nd column shows the mAP values for the "Moderate", and the 3rd column shows the mAP values for "Hard" objects. These categories are defined by the KITTI benchmark site.

**TABLE 1.** PointPillars parameters used for training.

| Parameter | Value |
|---|---|
| voxel_generator / point_cloud_range | [0, -39.68, -3, 69.12, 39.68, 1] |
| voxel_generator / voxel_size | [0.16, 0.16, 4] |
| voxel_feature_extractor / num_features | [64] |
| train_input_reader / batch_size | 4 |
| train_input_reader / prefetch_size | 25 |
| train_input_reader / max_number_of_voxels | 12000 |
| initial_learning_rate | 0.0002 |
| decay_steps | 27840 |
| decay_factor | 0.8 |

### 2) THE 1P25P(N) NETWORK

Second approach a vector of 25 pixel intensities is attached to each lidar point $\mathbf{P}_i$. Let us denote this vector by $\mathbf{v}_i$, which contains the intensity values of the $5 \times 5$ sized neighborhood of the projection $\mathbf{p}_i$. Let us denote this neighborhood by $\mathbf{U}_i$. $\mathbf{v}_i$ can be expressed as $\mathbf{v}_i = Vec(\mathbf{U}_i)$. In order to ensure accurate comparison across $\mathbf{U}_i$ we normalized the elements of $\mathbf{U}_i$ to have zero-mean and unit-variance (see Fig.5). However we have also tested the case when non-normalized neighborhood intensities are used for augmentation (see Fig. 4). By including neighborhood related information to the features of each 3D point, the network during training may ''utilize'' spatial image information, as well.

## V. RESULTS

### A. EVALUATION OF RESULTS ON A SEPARATED TEST SET

The performance of the detectors was tested on a separated test set containing 1000 training images from the KITTI 3D Benchmark. The metrics used for comparison here are the precision, recall and the mean Average Precision (mAP). The latter is calculated by averaging AP values over multiple Intersection over Union (IoU) thresholds used by COCO [28], [29].

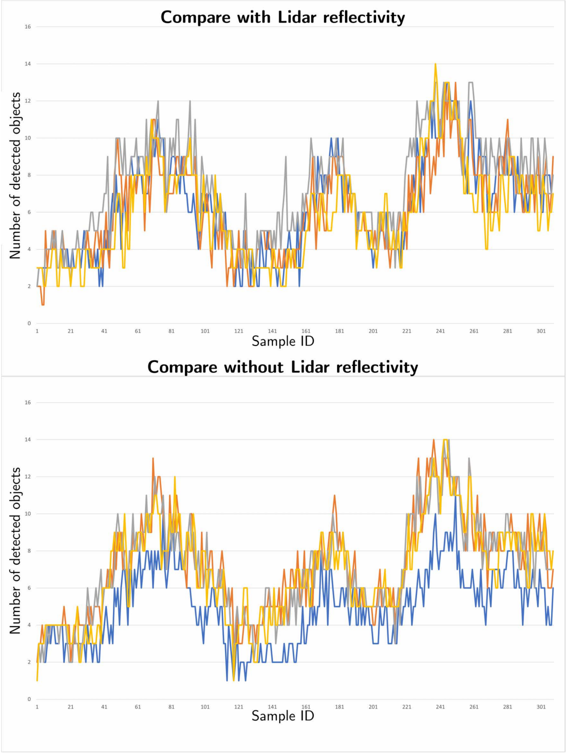


**FIGURE 7.** Number of detected objects on the same KITTI raw data scenario, with (top) and without lidar reflectivity values (bottom).

Two groups of detectors (each using the same baseline model but different augmentation) were compared. The

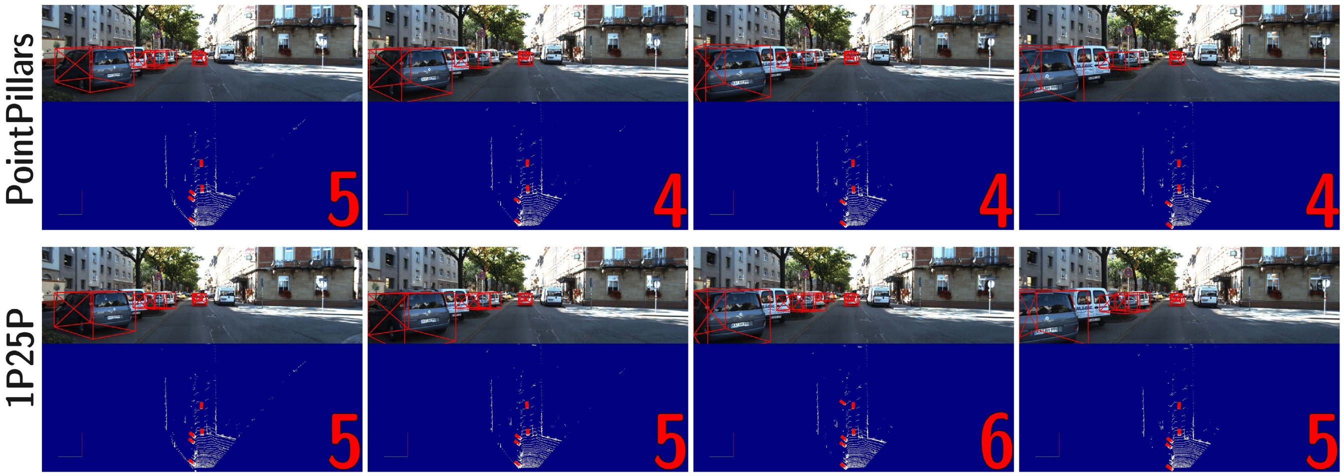


**FIGURE 8.** Detection results on the first sequence of the KITTI raw scene. First row: PointPillars, second row: 1P25P architecture. Lidar reflectivity values are also included.

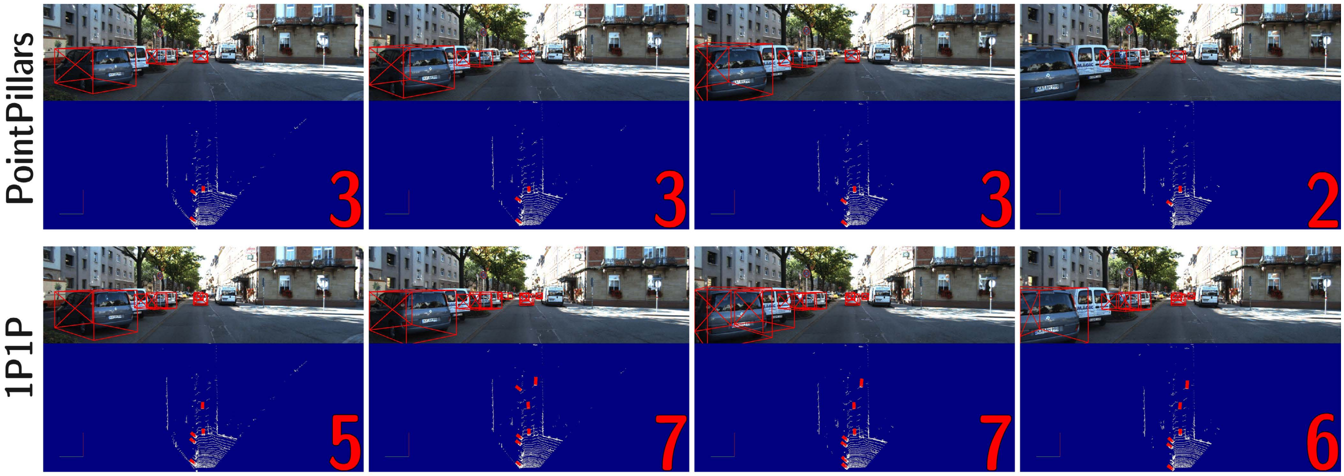


**FIGURE 9.** Detection results on the first sequence of the KITTI raw scene. First row: PointPillars, second row: 1P1P architecture. The lidar reflectivity values are omitted.

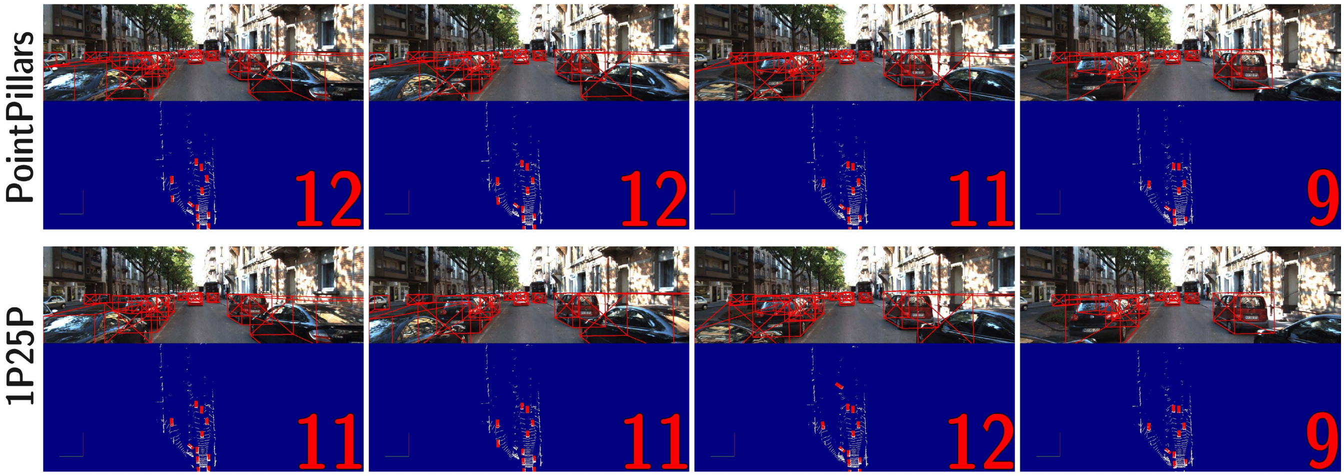


**FIGURE 10.** Detection results on the second sequence of the KITTI raw scene. First row: PointPillars, second row: 1P25P architecture. Lidar reflectivity values are also included.

first group uses all data from the lidar sensor, i.e. the pointcloud as well as the reflectance value for each lidar point. In the second group of networks the reflectance was omitted in order to eliminate the influence of differ-

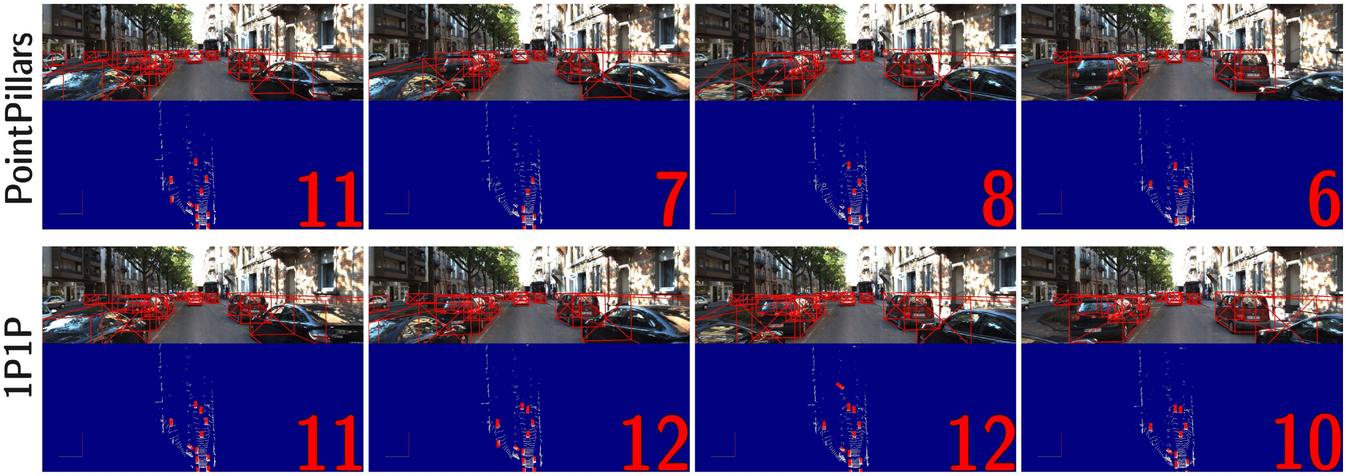


**FIGURE 11.** Detection results corresponding to the second sequence of the KITTI raw scene. First row: PointPillars, second row: 1P1P architecture. Lidar reflectivity values are omitted.

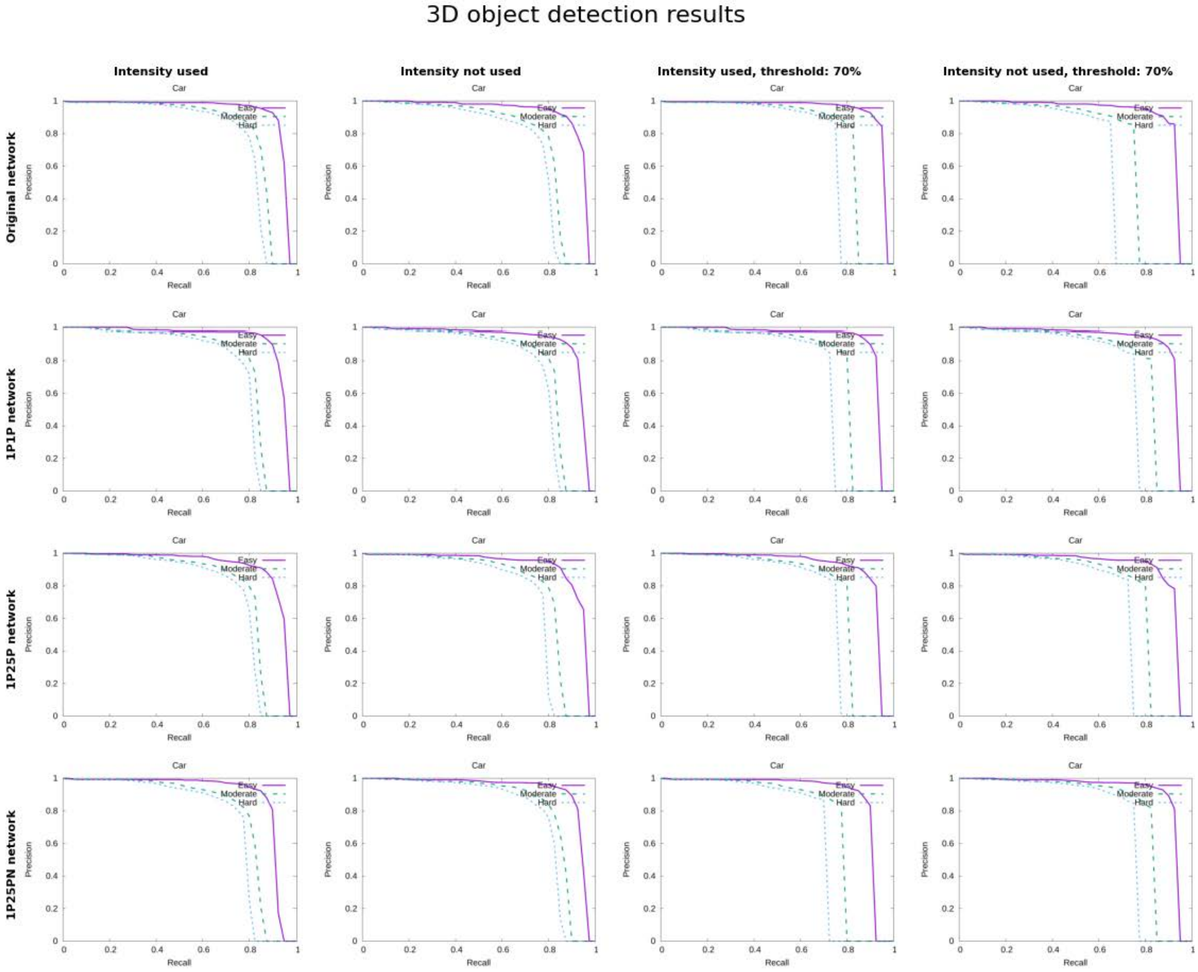


**FIGURE 12.** KITTI 3D object detection evaluation metric for each network architecture. The individual rows depict the recall-precision curves for the original PointPillars, the 1P1P, the 1P25P and the 1P25PN networks, respectively. The 1st column corresponds to the recall-precision curve for the case when the lidar reflectivity was also considered while the 2nd column reflect the case when the lidar reflectivity was omitted. The 3rd and 4th column correspond to cases when the detection threshold was set to 70% with lidar reflectivity included and omitted, respectively.

ent reflectance encoding schemes being used across lidar manufacturers. Obviously in this case the network is forced to learn from reduced data however our goal here is to substitute the reflectance value by image pixel intensities

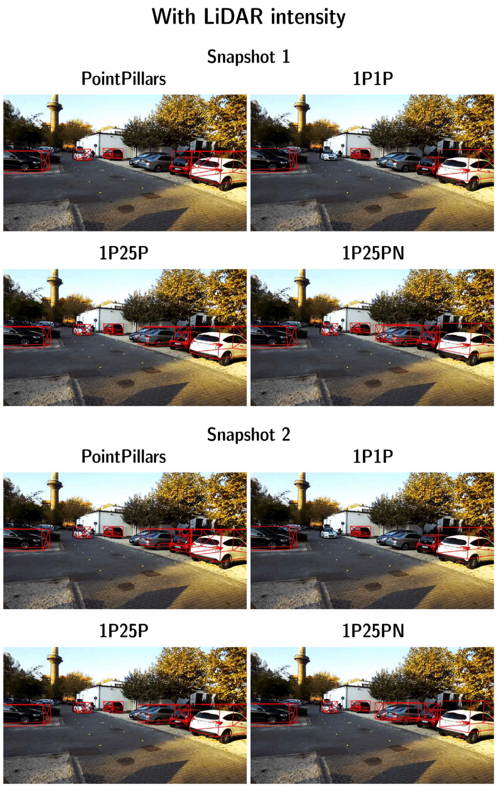


**FIGURE 13.** Cars detected in two snapshots from our custom dataset. Lidar reflectivity was also included. The 1st column of the 1st row shows the detections resulted by the original PointPillars network, the 2nd column of the 1st row reflects the results corresponding to the 1P1P network. In the 1st column of the 2nd row the detections obtained by the 1P25P network can be followed. The 2nd column of the 2nd row shows the detections resulted by the 1P25PN network.

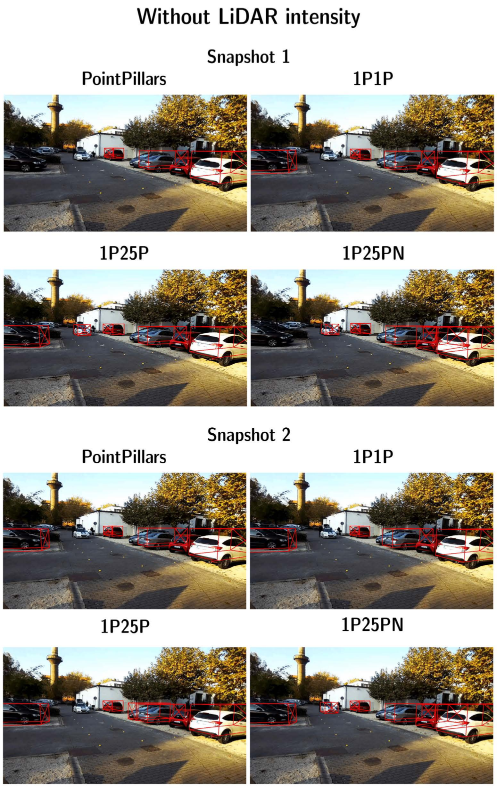


**FIGURE 14.** Cars detected in two snapshots from our custom dataset. Here the lidar reflectivity was omitted. The columns of the 1st row show the detections resulted by the original PointPillars and 1P1P network, respectively. The columns of the 2nd row show the detections resulted by the 1P25P and the 1P25PN networks, respectively.

and show their impact on the performance of trained detectors.

During training, the weights for all considered networks were saved at every 5000 steps for which the mAP metric was calculated on the test set (with available groundtruth) for each category (easy, moderate, and hard) according to the KITTI benchmark site [13](see Fig. 6). One can see that in case when the reflectance is included, the image based augmentation has no remarkable effect on the mAP (less than 1% difference). On the other hand when the reflectance is omitted, the image augmentation caused observable increase in the mAP. The largest contribution of image pixel intensities to mAP improvement can be observed in case of hard objects, i.e. when the number of rays reflected from the surface of objects is small.

The training of detectors was stopped after a certain number of steps which in case of the original and 1P1P detectors was roughly 300000 steps while in case of the 1P25P(N) networks roughly 600000 steps. We expected that more steps will be required for training a more complex network, but none of the considered networks produced remarkable improvement after 300000 steps. Each network was trained on the same splits of the KITTI dataset. To train the models, the hyperparameters used by the baseline model have been considered. The values of the most relevant hyperparameters are given in Table 1.

## B. TEST RESULTS ON KITTI RAW DATA SCENARIO

In this section, we selected the weights from the detectors that performed best in the evaluation process. Depending on whether the reflectance was included or omitted the 1P1P and 1P25P networks showed the best performance (see Figs.12 and 6). The selected weights were used to run through the network the "0104" drive data from the KITTI RAW dataset recorded on 26.09.2011 [23]. Although due to the absence of groundtruth data, the previously applied metrics were not calculated here, Figs. 8–11 reflect a remarkable improvement in the detector's performance.

In Fig. 8 a sequence of 5 frames can be seen. The top row shows the detections resulted by the original architecture while the bottom row shows the detections obtained by the 1P25P network. Here the lidar reflectivity values have also been taken into account. One may observe that there is no significant difference between the number of the detected objects for this sequence, thus the contribution of image features to the overall detection performance is negligible in this particular case.

Fig. 9 shows the same sequence of 4 frames. The top row shows the detections of the original architecture and the

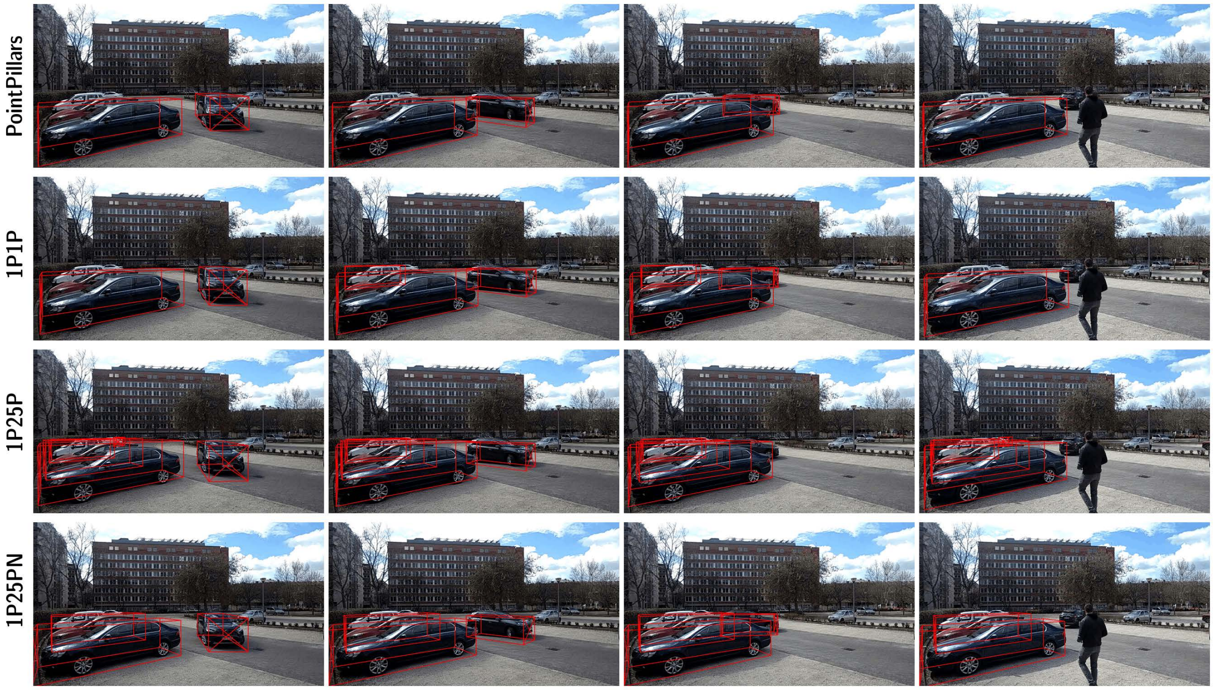


**FIGURE 15.** Detected vehicles in our custom dataset. The lidar reflectivity values were also included. The same four samples were given as input to the networks to compare their responses. The individual rows correspond to the responses of the original PointPillars, the 1P1P, the 1P25P and the 1P25PN networks, respectively.

bottom row shows the detections of the 1P1P network. In this series, the lidar reflectivity values were omitted from both the original as well as from 1P1P network. As one may observe the 1P1P network was able to detect more distant or occluded cars with a confidence larger than 70%, thus in this case the contribution of image features to performance improvement is remarkable.

Figure 10 shows another sequence, also consisting of 5 frames. The top row shows the detections of the original architecture and the bottom row shows the detections of 1P25P network. In this series, the lidar reflectivity values have been taken into account. There is no remarkable difference between the performance of the two networks. The same set of vehicles is detected by both networks, even in terms of orientation and location accuracy they are nearly of the same quality. Thus, by using image pixel intensity besides the lidar reflectivity, the performance improvement of the network is negligible.

In Fig. 11 the top row shows the detections resulted by the original architecture while the bottom one shows the detections yielded by the 1P1P network. Here the lidar reflectivity values were omitted. The number of objects detected by the 1P1P network compared to the original one (with reflectance omitted) has increased significantly. The confidence limit was set to 70%. The original network was able to detect nearby vehicles confidently, but in case of distant or occluded cars it did not perform as reliably as the 1P1P network did with image features. Significant increase in the number of true positive detections can be observed in case of the 1P1P network.

This section showed a comparison of the original and the 1P1P network for the case when lidar reflectivity values were taken into account, and the 1P25P network for case when the reflectivity was omitted. Here we considered these two networks only because according to Fig. 12 they proved to perform remarkably better than 1P25PN.

### C. DETECTOR PERFORMANCE ON OUR CUSTOM RECORDED DATA

We recorded our custom dataset with a different type of lidar sensor and camera than the one used by the KITTI vision benchmark suite. From numerous recordings, two groups were selected to test the contribution of image features on the detection performance. The confidence limits of detections were set to 70% and 75%. The networks were tested on the same snapshots.

#### 1) THE FIRST SCENARIO FROM OUR CUSTOM DATASET

Fig. 13 shows the detections when the lidar reflectivity was taken into account on the same short series of recordings. The confidence limits of the detections were set to 75%. The results show no difference in the number of detected objects in this case. There are some cases where the detectors (original, 1P1P, 1P25P, 1P25PN) recognize different vehicles but the overall performance has not been improved.

The detectors were also tested on these recordings by omitting the lidar reflectivity (see Fig. 14). The results show that the modified networks detect more vehicles on these frames. The reason behind this might be the

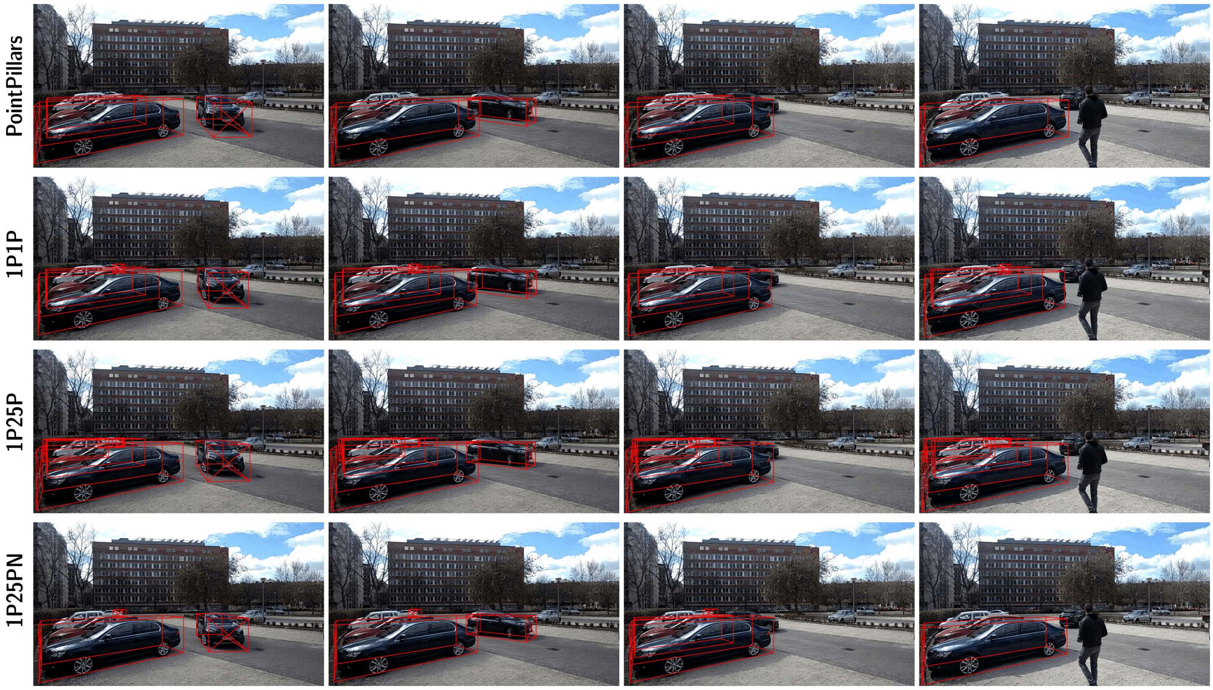


**FIGURE 16.** Detected vehicles in our custom dataset. Lidar reflectivity values were not used by the networks in this case. The same four samples were given to the networks to compare their responses. The individual rows correspond to the responses of the original PointPillars, the 1P1P, the 1P25P and the 1P25PN networks, respectively.

low number of points for each vehicle due to occlusions. As it can be seen the 1P25P and the 1P25PN networks detected most of the vehicles, while the original network (by omitting the lidar reflectance) provided fewer detentions. Neither of the detectors was able to detect all vehicles in this scene.

#### 2) THE SECOND SCENARIO FROM OUR CUSTOM DATASET

Similarly to the first scenario, the detectors were evaluated with and without considering the lidar reflectance (see Fig. 15) and Fig. 16). The same phenomenon can be observed as in case of the first scenario. By including the reflectivity values the performance did not change. On the other hand by omitting reflectivity values the modified architecture proved to be more effective.

### D. HARDWARE SETUP AND CALIBRATION

For recording the stream of image-pointcloud pairs a Hikvision DS-2CD2063G0-I camera having 6MP resolution and an Ouster OS-1 Uniform 64 channel lidar sensor was used. The calibration of the camera was performed by the method proposed by Zhang [30]. The Camera-lidar extrinsics have been estimated by the method proposed in [27].

The detector works by projecting the lidar points onto the camera plane, thus in addition to an accurate calibration, it is essential to precisely synchronize the acquisition of data in order to determine the correct pixel intensity value corresponding to a given 3D point. The importance of time synchronisation is illustrated by Liu et. al. in Matter of time [31]. Inaccurate synchronization may affect the performance of the detector significantly.

## VI. CONCLUSION

Reliable environment sensing is one of the most important tasks for self-driving vehicles. The most common types of available object detectors are the lidar only, camera-only and the camera-lidar based detectors. In this paper a low level camera-lidar fusion was proposed based on augmentation of pointcloud data by image features to improve the performance of lidar only based detectors. It was shown how pixel intensity patterns (compared to 3D spatial data) contribute to the reliability of detections especially in those cases when distant objects (represented by lower number of points in the pointcloud) have to be detected. The augmentation is performed by attaching reshaped image intensity patterns to each projected 3D point in the pointcloud. The network retains 20 FPS, which corresponds to the highest frame rate of available lidar sensors. The accuracy of the detector was evaluated and tested on the KITTI dataset as well as on custom data.

## ACKNOWLEDGMENT

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

## REFERENCES

[1] H. Zhu, J. Deng, Y. Zhang, J. Ji, Q. Mao, H. Li, and A. Zhang, "VPFNet: Improving 3D object detection with virtual point based LiDAR and stereo data fusion," 2021, *arXiv:2111.14382*.

[2] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4604–4612.

[3] X. Dong, B. Zhuang, Y. Mao, and L. Liu, "Radar camera fusion via representation learning in autonomous driving," *CoRR*, vol. abs/2103.07825, pp. 1–10, Jun. 2021.

[4] Z. Zhong, S. Liu, M. Mathew, and A. Dubey, "Camera radar fusion for increased reliability in ADAS applications," *Electron. Imag.*, vol. 30, no. 17, pp. 258-1–258-4, Jan. 2018.

[5] H. Hajri and M.-C. Rahal, "Real time LiDAR and radar high-level fusion for obstacle detection and tracking with evaluation on a ground truth," *CoRR*, vol. abs/1807.11264, pp. 1–7, Jul. 2018.

[6] R. Roriz, J. Cabral, and T. Gomes, "Automotive LiDAR technology: A survey," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 15, 2021, doi: 10.1109/TITS.2021.3086804.

[7] G. Ágoston and R. Madleňák, "Road safety macro assessment model: Case study for Hungary," *Periodica Polytechnica Transp. Eng.*, vol. 49, no. 1, pp. 89–92, Jan. 2020.

[8] V. Tihanyi, A. Rövid, V. Remeli, Z. Vincze, M. Csonthó, Z. Pethő, M. Szalai, B. Varga, A. Khalil, and Z. Szalay, "Towards cooperative perception services for ITS: Digital twin in the automotive edge cloud," *Energies*, vol. 14, no. 18, p. 5930, Sep. 2021.

[9] C. R. Qi, H. Su, K. Mo, and J. L. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *CoRR*, vol. abs/1612.00593, pp. 1–19, Dec. 2016.

[10] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," *CoRR*, vol. abs/1711.06396, pp. 1–10, Nov. 2017.

[11] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[14] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-ensembling single-stage object detector from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2021, pp. 14494–14503.

[15] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov, "SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15446–15456.

[16] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Computer Vision—ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Ed. Cham, Switzerland: Springer, 2020, pp. 35–52.

[17] Q. Xu, Y. Zhong, and U. Neumann, "Behind the curtain: Learning occluded shapes for 3D object detection," 2021, *arXiv:2112.02205*.

[18] S. Pang, D. Morris, and H. Radha, "CLOCs: Camera-LiDAR object candidates fusion for 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10386–10393.

[19] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.

[20] S. Azam, F. Munir, A. M. Sheri, Y. Ko, M. Hussain, and A. Jeon, "Data fusion of LiDAR and thermal camera for autonomous driving," in *Proc. Appl. Ind. Opt., Spectrosc., Imag. Metrol.*, Jan. 2019, pp. 1–3.

[21] P. Fritsche, B. Zeise, P. Hemme, and B. Wagner, "Fusion of radar, LiDAR and thermal information for hazard detection in low visibility environments," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 96–101.

[22] D. Gohring, M. Wang, M. Schnurmacher, and T. Ganjineh, "Radar/LiDAR sensor fusion for car-following on highways," in *Proc. 5th Int. Conf. Autom., Robot. Appl.*, Dec. 2011, pp. 407–412.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[24] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, "A multimodality fusion deep neural network and safety test strategy for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 310–322, Jun. 2021.

[25] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, Feb. 2020.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, E. S. Reed, C. Fu, and C. A. Berg, "SSD: Single shot MultiBox detector," *CoRR*, vol. abs/1512.02325, pp. 1–17, Dec. 2015.

[27] W. Wang, K. Sakurada, and N. Kawaguchi, "Reflectance intensity assisted automatic and accurate extrinsic calibration of 3D LiDAR and panoramic camera using a printed chessboard," *Remote Sens.*, vol. 9, no. 8, p. 851, Aug. 2017.

[28] *Detection Evaluation Metrics Used by COCO*. Accessed: Jan. 26, 2022. [Online]. Available: https://cocodataset.org/#detection-eval

[29] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *CoRR*, vol. abs/1405.0312, pp. 1–15, May 2014.

[30] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Dec. 2000.

[31] S. Liu, B. Yu, Y. Liu, K. Zhang, Y. Qiao, T. Y. Li, J. Tang, and Y. Zhu, "The matter of time—A general and efficient system for precise sensor synchronization in robotic computing," *CoRR*, vol. abs/2103.16045, pp. 1–4, Mar. 2021.

**MIHÁLY CSONTHÓ** was born in Dunaújváros, Hungary, in 1996. He received the M.Sc. degree in autonomous vehicle control engineering from the Budapest University of Technology and Economics, Hungary, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering.

He is also working as a Research Assistant at the BME Automated Drive Laboratory. His main research interests include developing perception algorithms for autonomous vehicles and for infrastructures, and sensor fusion.

**ANDRÁS RÖVID** was born in Rimaszombat, Slovakia, in 1978. He received the degree in computer engineering from the Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Kosice, Slovakia, in 2001, and the Ph.D. degree in transportation sciences from the Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 2005. Since 2019, he has been the Leader of the Perception Group, Department of Automotive Technologies, BUTE, where he is currently a Senior Research Fellow. He is the author or coauthor of over 100 publications. His main research interests include image processing, 3D machine vision, and sensor fusion.

**ZSOLT SZALAY** received the M.Sc. degree in electrical engineering from the Budapest University of Technology and Economics (BME), Hungary, in 1995, the M.Sc. degree in business administration from Corvinus University, in 1997, and the Ph.D. degree in mechanical engineering from the BME, in 2002. He is currently an Associate Professor and the Head of the Department of Automotive Technologies, BME. He also acts as the Head of Research and Innovation with ZalaZONE Automotive Proving Ground, the unique Hungarian infrastructure for connected and automated vehicle testing. His research interests include advanced automotive technologies related to the testing and validation of highly automated and autonomous vehicles. He is a Committed Supporter of young talents from an early age as a Children's University lecturer and via the BME Automated Drive Laboratory.