2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

The TUBS Road User Dataset: A New LiDAR Dataset and its Application to CNN-based Road User Classification for Automated Vehicles

Christopher Plachetka, Jens Rieken and Markus Maurer
Institute of Control Engineering
Technische Universität Braunschweig
Braunschweig, Germany
Email: {plachetka,rieken,maurer}@ifr.ing.tu-bs.de

***Abstract*—We present a novel approach for classifying pre-segmented laser scans of road users with consideration of real-time capability for applications in automated vehicles. Our classification approach uses 2.5D Convolutional Neural Networks (CNNs) to process range data as well as intensity information retrieved from reflected beams. We do not solely rely on publicly available laser scan datasets, which lack several features, but we provide an additional dataset from real-world sensor recordings, annotated by a tracking-based automatic labeling process.**
We evaluate the classification performance of our CNN regarding different feature configurations. For training, we use automatically and manually labeled data as well as mixtures with other public datasets. The results show promising classification capabilities. Training with automated labels shows similar results, providing a possibility to avoid the need for manual editing expense.

I. Introduction

Automated driving has a huge potential to increase traffic safety for the driver himself and for other road users. Environment perception is a crucial task for automated vehicles, because it has to replace all tasks that were formerly accomplished by human visual perception, e. g. localizing within lanes, classifying other roads users or determining their velocities. Road user classification also impacts behavioral decisions.

Automated vehicles are not restricted to rely on visual imagery for environment perception only, but utilize various kinds of sensors. Among those are LiDAR-based sensors, which sample the environment by a large number of point-like distance measurements, providing range information in high density and resolution.

In order to process sensor data, Deep Learning is an alternative to conventional algorithms. Neural Networks are often applied to realize applications in the automotive domain [1]–[6].

In recent years, CNNs have shown impressive results in image classification tasks [7]–[9]. As a consequence, CNNs are applied to range data as well. Architectures that were designed for image data processing, can be utilized for this purpose [10]–[13]. Those are typically referred to as *2.5D CNNs*. Alternatively, three-dimensional convolution can be performed. However, the need for discretization and the additional dimension lead to increased computational costs, which leaves 2.5D CNNs as the more suitable approach for achieving real-time capability for road user classification tasks.

Within the project *Stadtpilot* [14], the TU Braunschweig develops an experimental automated vehicle that incorporates a Velodyne HDL-64E laser scanner for environment perception. The sensor provides dense range and intensity data, as shown in Fig. 1. To tackle the need for training data required for Deep Learning-based approaches, we utilize an already available object tracking system [15], [16] to automatically label road users in consecutive laser scans. Apart from large amounts of data generated by this approach, we manually annotate a small subset in order to provide a ground truth for validation and testing. This also serves as a dataset for training with manually labeled samples. The contributions of our work are summarized as follows:

- Provision of a new dataset, with labeled road users in the scanner's entire viewing range, containing 120 000 automatically labeled laser scans (≈1.5 mio. objects) distributed among seven classes.
- Provision of 850 manually labeled laser scans, along with the developed labeling tool.
- Road user classification using a 2.5D CNN approach.
- Demonstration of increased classification accuracy by additionally exploiting intensity data.
- Comparison of training results with manually and automatically labeled data, showing that the latter achieves comparable results if label noise is properly considered.

To the best of our knowledge, we are the first to provide a laser scan dataset labeled with road users in the scanner's entire field of view. Besides, we are the first to perform a detailed classification of road users using a 2.5D CNN that processes range and intensity data from a laser scanner.

978-1-7281-0323-5/18/$31.00 ©2018 IEEE

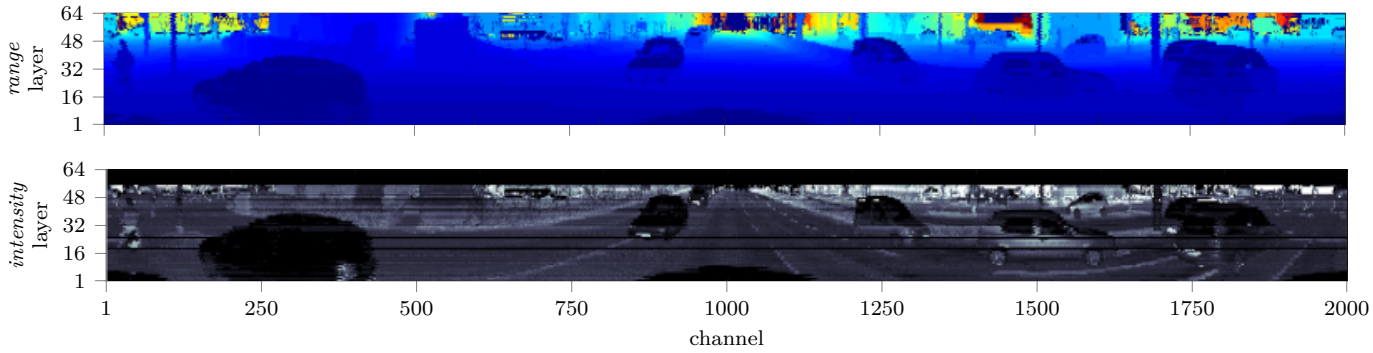

Fig. 1. Range and intensity values provided by the vehicle's laser scanner.

This paper is structured as follows. Sec. II reviews the state of the art of laser scan processing with CNNs in terms of road user classification. Sec. III briefly describes the properties and features of our dataset. Sec. IV gives insights into the design of the presented 2.5D CNN. Sec. V and VI discuss the results obtained by using manually and automatically labeled data, respectively.

II. Related Work

Approaches to process laser range data with CNNs can be divided into two categories. Conventional 2D architectures for CNN-based image processing can be utilized by including range measurements as additional input. 3D CNNs add an additional dimension and can thus be considered as an extension of 2D CNNs. Because of their high computational demands, they are not considered in the following.

It is known, that the utilization of spatial information as input to Neural Networks enhances classification performance, compared to solely using RGB data [17]. The first approaches that included spatial information in CNNs were using RGB-D images [10]–[13]. Herein, the range channel is handled differently: Eitel *et al.* [10] and Schwarz *et al.* [11] code range as RGB colors, which allows the CNN to benefit from already available, pre-trained architectures. The fact that range data has successfully been used without recoding it as RGB, as done by Socher *et al.* [12] and Alexandre [13], leads to the conclusion that CNNs are not limited to RGB data processing, but are able to extract features directly from range information.

A 2.5D CNN is utilized by Li *et al.* [18] to detect vehicles in a laser scan by producing objectness scores and bounding box proposals. The laser scan is coded in cylindrical coordinates using points' heights above ground and range information. For the same purpose, Chen *et al.* [19] use LiDAR data in multiple views (bird's eye and front view, including intensity values) and RGB data. Dewan *et al.* [20] perform a semantic classification of laser scans. Using a Fully Convolutional Network (FCN), each point of a scan is classified in either movable, stationary or moving. A. Zelener and I. Stamos [21] perform a segmentation of laser scans regarding vehicles and background. The scans are collected by Google Street View cars. Range, height and two angular channels are used as input for a CNN that labels patches of the laser scan.

A more detailed segmentation in cars, bicycles and pedestrians is obtained by Wu *et al.* [22]. They use a FCN with range and intensity data as input, which is constructed from 3D point clouds. Training data is enhanced by simulated data, improving the network's performance. Since those approaches use range, height and intensity values with a 2.5D CNN, they are the most related work to ours regarding object detection. In contrast to Wu *et al.*, we perform a more detailed classification of previously segmented objects instead of a segmentation and improve our network's performance by using height values.

Publicly available datasets providing 3D range data

In terms of dataset size and number of classes, only the widely-used KITTI Vision Benchmark Suite [23] is suitable for our demands regarding Deep Learning-based road user classification. However, the main drawback of the KITTI dataset is the lack of labels in the rearward half of the laser scan, due to the front camera-based labeling procedure. Half labeled scans constrain training procedures and model architectures, as presented by Li [24], where bounding box predictions can be made only for a scan's front half. Our dataset overcomes this drawback by providing labels in the scanner's entire field of view, since our labeling procedure is not camera-based, but labels point clouds directly.

III. The TUBS Road User Dataset

The following section describes the features and properties of our dataset. The set contains automatically and manually labeled laser scans, with labels for point cloud data as well as information about tracked objects. Both are available within the entire field of view of the sensor, as shown in Fig. 2. In addition, information about the host vehicle's movement is provided. In order to generate large amounts of automatically labeled data, we take advantage of an existing object tracking system. A subset is reviewed manually to provide more accurate ground truth information.

Each laser scan consists of 128 000 range and intensity measurements, respectively. They are organized in 64 layers and 2 000 channels, based on the sensor's resolution.

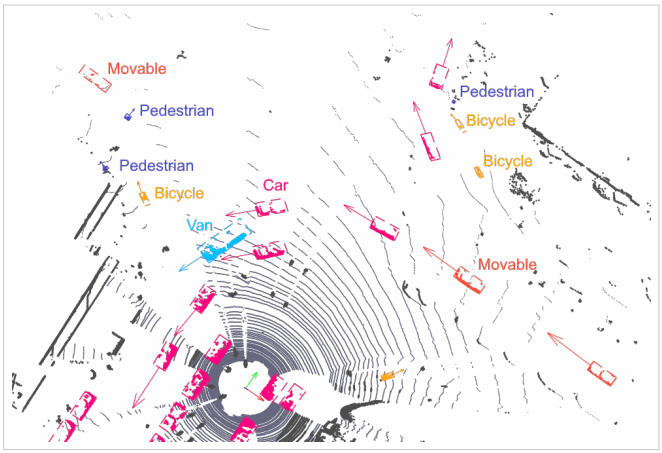

Fig. 2. Laser scan of the TUBS Road User Dataset, labeled with road users in the scanner's entire field of view.

Scans are provided with 10 Hz and in sequences of 50 consecutive scans, within which temporal consistency is ensured.

The training dataset contains 120 000 scans (≈1.5 mio. objects). A small manually labeled dataset (850 scans, ≈36 000 objects) is provided for validation and testing purposes. Currently the following labels are supported: Cars, vans, trucks, motorbikes, pedestrians, bicycles. A meta-class *movable* is available for all undefined, yet moving objects (e. g. if the class cannot be determined or is not encoded yet). All remaining points are labeled as stationary or part of the road surface. Information about tracked objects includes pose and extent of bounding boxes as well as dynamic properties and class labels.

IV. Road User Classification

The following section describes our road user classification approach based on a 2.5D CNN. Its architecture is presented, and details about the training procedure are addressed, such as normalization strategies and data representations. The classification process is performed on pre-segmented point clouds, which are provided by preprocessing stages of the mentioned object tracking system, and is done independently for each segment. Such a segment is denoted as *object* in the following.

A. CNN Architecture

For most experiments, a rather simple CNN architecture consisting of two convolutional (conv) and two fully connected (fc) layers is used, as shown in Fig. 3. In order to demonstrate the principle classification capability of a 2.5D approach, more complex architectures weren't chosen on purpose.

Input of the network can consist of up to three modalities[1], illustrated in Fig. 4. Both *range* and *height above ground* modalities provide geometric information, whereas the *intensity* modality provides information about the object's surface, namely its reflectivity for near-infrared radiation. The latter might be useful for the detection of high-reflective features like vehicle lights or license plates, but also for distinction of surface materials.

In order to fully cover objects even close to the scanner, crops (explained below) are required to contain up to 400 columns of the laser scan. This leads to a high overall number of neurons (1.6 mio., 100.4 mio. parameters).

B. Sample generation and performance metrics

Training, validation and test samples were generated by cropping objects out of labeled laser scans. Samples for the *stationary* class were extracted algorithmically by randomly taking crops that are not close to any movable object. We manually labeled 850 laser scans and split them into validation (450) and testing sets (400), resulting in 36 217 manually labeled objects in total. Table I shows the class distribution of these datasets as well as for the automatically generated training dataset.

Classification performance as well as the ability to separate stationary from movable objects are in focus of this paper. The latter, referred to as detection performance, is evaluated as a two-class-problem (movable vs. stationary) by merging all different object classes but the class *stationary*.

For the detection performance, precision, recall, f_1-score, False Positive Rate (FPR) and True Negative Rate (TNR) metrics are applied. The classification performance is evaluated using:

- Top-1 accuracy $\mathrm{acc}_{\mathrm{Set}}$: Ratio of all correctly to all falsely classified objects in the dataset.
- Mean class accuracy $\mathrm{acc}_{\mathrm{CM}}$: The arithmetic mean of all class-specific accuracies.

C. Training procedure

To train our CNN, we applied a weighted cross entropy loss to balance the dataset in order to make use of all available samples. The cost function $\mathcal{L}$ is given in Eq. 1 and Eq. 2, where N_B denotes the batch size, K the number of classes, w_k a class specific weight and $\boldsymbol{x}^{(i)}$ a laser scan crop. P_T and P_O denote the target's probability for class k and the output probability, respectively. The weight w_k is given by the ratio of the total amount of samples N_T in the dataset to the class specific amount $N_C\,(k)$ in this dataset, amplifying the loss caused by less common classes in the batch.

$$\mathcal{L} = \frac{-1}{N_B} \sum_{i=1}^{N_B} \sum_{k=1}^{K} w_k \cdot P_T(k \mid \boldsymbol{x}^{(i)}) \cdot \log\left(P_O(k \mid \boldsymbol{x}^{(i)})\right) \tag{1}$$

$$w_k = \frac{N_T}{N_C\,(k)} \tag{2}$$

For all our experiments we used a learning rate of 1×10^{-3} and a batch size of 256. We only considered objects within 60 m distance because of declining point density.

[1]The term *modality* is introduced here to provide a distinction from the term *channel*. Although from the network's point of view modalities are treated as (image-like 2D) input channels, we would like to put emphasize on the different information represented by the mentioned types of data.

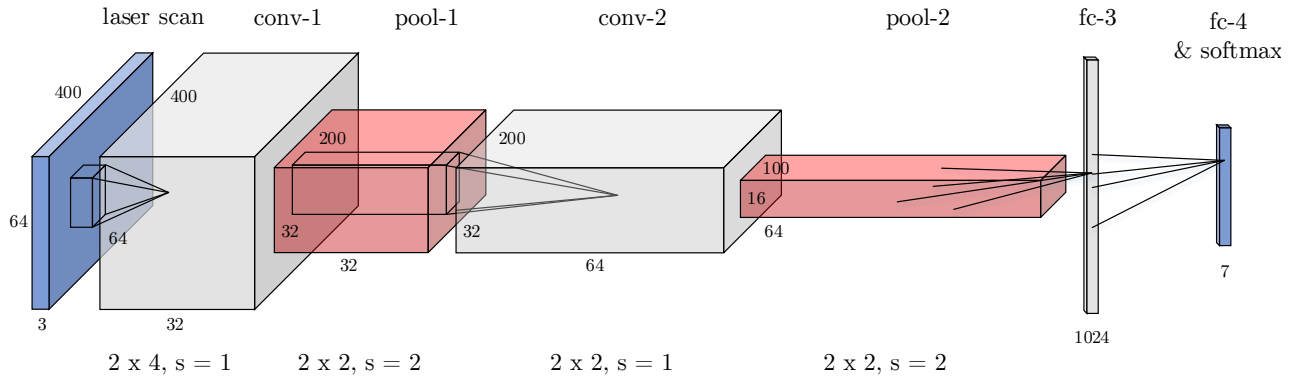

Fig. 3. Architecture of the proposed 2.5D CNN. 64×400 (row $\times$ column)-sized crops centered at each processed segment are utilized as input. The crop's depth corresponds to the number of input modalities, which differs in our experiments (range, intensity, height above ground and combinations). The crop is then convolved with 32 filters using a 2×4 receptive field with the stride $s = 1$. The second conv-layer uses 64 symmetric 2×2 filters. Between layers we placed 2×2 pooling layers. The convolution layers are followed by two fc-layers with 1024 and seven neurons (corresponding to seven classes). The last fc-layer has a softmax activation to output class probabilities.

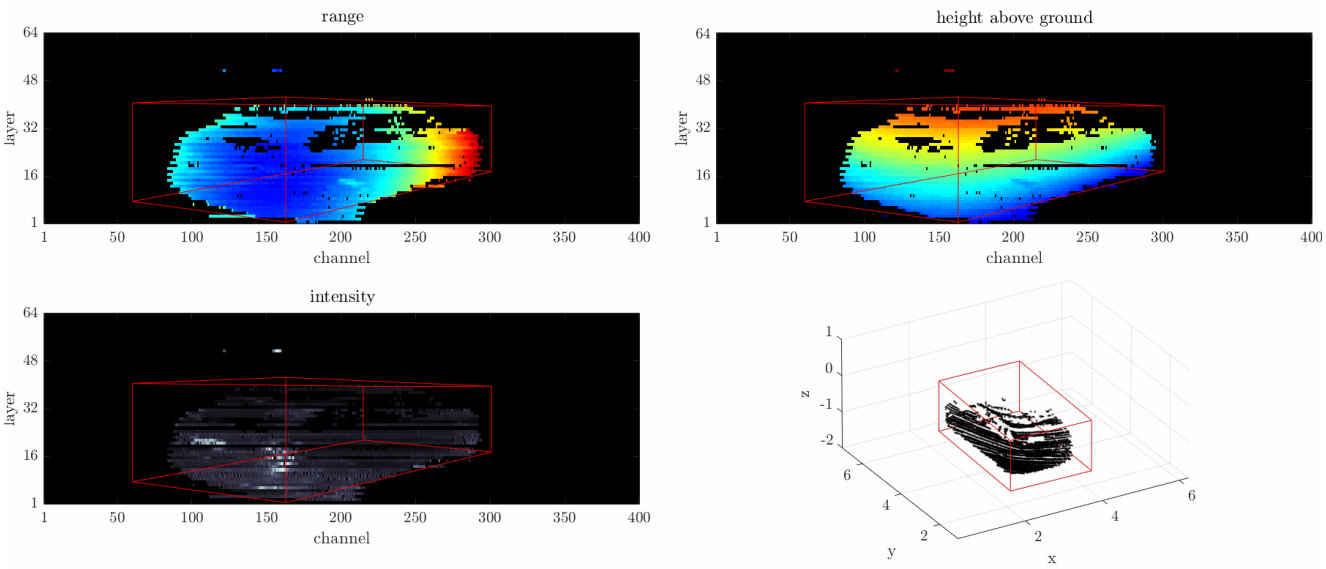

Fig. 4. Top left: Range channel of an object's laser scan crop. Brighter colors indicate greater values in all images. Top right: Height above ground channel, which corresponds to the z-coordinates of points in the Cartesian coordinate system that originates in the scanner's center. Bottom left: Intensity channel. Bottom right: Corresponding point cloud.

TABLE I
DISTRIBUTION OF SAMPLES OVER CLASSES WITHIN DIFFERENT DATASETS

	movable classes												stationary class	
	car		van		truck		motorbike		bicycle		pedestrian			
	abs.	rel. (%)	abs.	rel. (%)	abs.	rel. (%)	abs.	rel. (%)	abs.	rel. (%)	abs.	rel. (%)	abs.	rel. (%)
a)	7 187	38.6	727	3.9	485	2.6	305	1.6	733	3.9	1 079	5.8	8 110	43.5
b)	6 957	39.2	978	5.5	389	2.2	173	1.0	1 759	9.9	545	3.1	6 957	39.2
c)	525 472	35.1	187 424	12.5	55 909	3.7	64 048	4.3	71 385	4.8	67 299	4.5	525 322	35.1
d)	19 048	42.7	2 777	6.2	1 037	2.3	(N/A)	(N/A)	1 557	3.5	3 953	8.9	16 194	36.3

Row a) and b) correspond to the manually edited validation and test dataset, respectively. Automatically generated samples correspond to row c). Row d) shows the class distribution of the dataset created from the KITTI Vision Benchmark Suite.

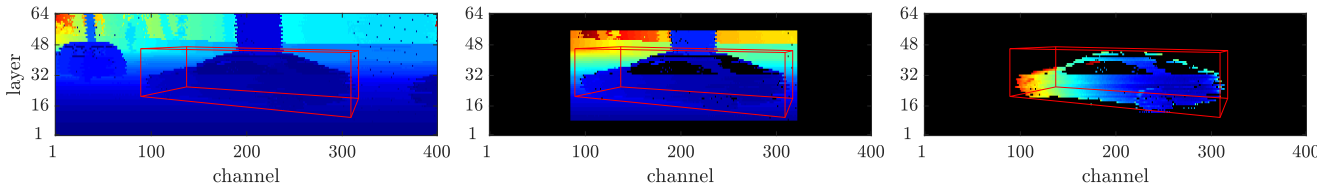

Fig. 5. Overview of considered data representation approaches, illustrated by the *range* modality. Left: Plain 64×400-sized crop. Center: *Bounding box crop* with a 10×10 margin. Note the advertising pillar in the car's background, which provides contradicting features for the class *stationary*. Right: 2.5D sparse scan. Here, only points within the 3D bounding box remain for network input.

Weights are initialized with $\sqrt{2/N}$, with N denoting the number of neurons, offsets are initialized with zero (cf. He *et al.* [25]). Furthermore, we applied weight decay ($\alpha = 5 \times 10^{-4}$) to all layers and dropout in the fc-1 layer, if not stated otherwise. Training was performed using TensorFlow™ with stochastic gradient decent without momentum.

D. Data representation and augmentation

Although 400×64-sized crops allow for full coverage of close objects, they also might contain additional objects with different than the annotated classes, with negative impact on classification results. By omitting all data outside an object's 2D bounding box, which is obtained by projecting its 3D bounding box onto an depth image-like representation of the scan, this issue can be solved. This representation is referred to as *bounding box crop*. Its application more than doubled the mean class accuracy acc_{CM} (from ≈0.3 to 0.7 in early experiments). In addition, using only points within the 3D bounding box was considered, denoted as *2.5D sparse scan*. Fig. 5 gives an overview of these representations.

When using the *2.5D sparse scan* representation, the application of dropout in the fc-1 or any other conv-layer led to a decreasing classification performance. We traced this effect to the sparse input tensor: Applying dropout leads to reduction of vital geometric information. Because dropout does not reduce classification performance when using *bounding box crop*, we conclude that the CNN treats the laser scan as a (focused) image in this case, although geometric information is partly being lost. On the other hand, using *2.5D sparse scan* representation without any background information supports the CNN to evaluate geometric information rather than treating the scan as an image.

In order to reduce overfitting, we did not take smaller crops for dataset augmentation, as widely done in image classification. Instead, the scan is shifted to the left or right by a random amount of channels. This equals a simple rotation of the laser scanner itself. Vertical shifting violates the geometric constrains; thus, it is not applied.

E. Normalization strategy

Different normalization strategies were considered. Best results were obtained using Global Contrast Normalization (GCN), applied independently to each input modality c, as given in Eq. 3 and 4.

$$\overline{S}_c = \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} S(i,j,c) \tag{3}$$

$$\widetilde{S}(i,j,c) = \frac{S(i,j,c) - \overline{S}_c}{\frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} \left(S(i,j,c) - \overline{S}_c \right)^2} \tag{4}$$

V. Results using manually labeled data

In the following we present our results regarding the training with manually labeled data. This dataset is utilized to train our network in order to identify each modality's effect and the best-performing input modality configuration. This setup is then used to identify the global best-performing setup, regarding various data representations and normalization strategies. Additionally, we evaluate our approach on a dataset generated from the KITTI Vision Benchmark Suite.

We used the comparatively small validation data set for training, thus omitting online validation. Our CNN is trained using 2 000 iteration steps for modality setup experiments and 3 000 steps for determining the best overall setup. A wide variety of experiments was conducted, including altering crop sizes, initialization and augmentation strategies, normalizations, usage of dropout and batch normalization, as well as different context margins for *bounding box crop* representation.

1) Investigating input modalities: For the following experiments we chose the *2.5D sparse scan* as data representation and applied different configurations of the available input modalities. Results are given in Tab. II. As expected, the usage of all three available modalities performs best regarding classification and detection performance. Note that using the range modality performs better than solely using the intensity modality.

TABLE II
Results for input modality experiments

	acc_{Set}	acc_{CM}	precision	recall	f_1	FPR	TNR
a)	0.74	0.68	0.80	0.94	0.87	0.35	0.65
b)	0.70	0.64	0.82	0.88	0.85	0.29	0.71
c)	0.83	0.71	0.89	0.98	0.93	0.18	0.82
d)	**0.83**	**0.74**	0.92	0.96	0.94	**0.13**	0.87

a) Range. b) Intensity. c) Range and intensity. d) Range, intensity and height above ground. Colored values indicate best achieved results. (acc_{Set}: Top-1 accuracy; acc_{CM}: Mean Class accuracy; FPR: False Positive Rate; TNR: True Negative Rate)

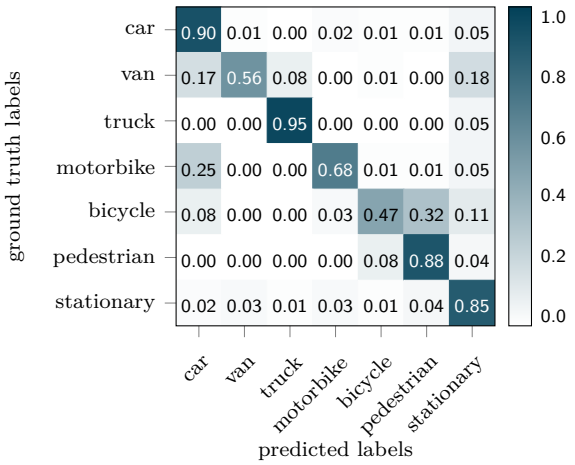

Fig. 6. Confusion matrix for the best-performing global setup

2) Best-performing global setup: In the following, all three input modalities were employed. In the global setup experiments, different configurations of normalization and data representation were applied. The highest acc_{Set} score was achieved in experiment a) with the usage of *bounding box crop* as data representation, with symmetric margin of 10 steps. Using the same data representation in experiment b), but omitting the GCN of the height channel, yielded the best detection performance (recall vs. FPR). The highest acc_{CM} was achieved in experiment c), in which we employed *2.5D sparse scan* as representation and GCN for each channel. The results are shown in Table III.

Fig. 6 shows the confusion matrix for experiment c). Highest classification rates were achieved for cars (0.90), trucks (0.95) and pedestrians (0.88). Bicycles and pedestrians get confused easily due to occlusions. The classification of an object into car or van is sometimes imprecise, even when the classification is done by humans. The high confusion of vans with stationary objects is likely explained by occluded vans due to their rectangular shape that resembles walls. A qualitative analysis of falsely classified vans supports this theory. The reason for the confusion of motorbikes and cars is originated in our small test set: It contains only one scooter that is labeled as a motorbike, which is always falsely classified. This leads to an over-proportional influence, given that motorbikes are the scarcest class.

We expect the classification results for our CNN to improve if we would use more samples for testing. Currently, many crops are different poses of the same object and, therefore, are correlated. This leads to a biased confusion matrix. A categorization of samples in multiple levels of difficulty regarding occlusion is planned to fan out classification results. Furthermore, we expect that the classification performance of 2.5D CNNs can be improved by using a deeper architecture while shrinking the input size. This could be done by down-sampling close objects to limit the number of parameters and to tackle scale variance.

3) KITTI dataset evaluation: In order to prove the capability of our approach to cope with other datasets, we constructed another dataset using point clouds from the KITTI Vision Benchmark Suite [23] by sampling 3D point clouds into 2.5D range and intensity images. Tab. I, row d) shows the resulting class distribution. Note that the KITTI dataset does not contain a motorbike class.

Tab. IV summarizes our results regarding the achieved performance using different dataset combinations for training and testing. First, we split the KITTI dataset in 60 % training and 40 % test samples. The good performance regarding acc_{Set} and acc_{CM} shows that our approach generally extends to other data sources. The increase in performance compared to applying the only TUBS datasets for training and testing is explained by the correlation of objects in the TUBS dataset resulting from consecutive point cloud frames. KITTI dataset currently contains more uncorrelated data.

Next, we applied the TUBS and KITTI dataset for training and evaluation, respectively, and vice versa. Both experiments reveal a bad generalization from one dataset to another, indicating differences between the datasets within same object classes. Most likely, these differences result from ground points that are included in KITTI objects, whereas ground points are filtered in TUBS objects. In addition, geometric inconsistencies (e. g. missing points) can occur due to the sampling method used to construct the 2.5D KITTI dataset from 3D point clouds. Nonetheless, the CNN is able to model those differences if the datasets are combined as in experiment d). Results are illustrated in Fig. 7. Samples were selected randomly and equally from both datasets

TABLE III
RESULTS FOR BEST-PERFORMING GLOBAL SETUP EXPERIMENTS

	acc_{Set}	acc_{CM}	precision	recall	f_1	FPR	TNR
a)	**0.86**	0.72	0.96	0.97	0.96	0.07	0.93
b)	0.85	0.72	0.96	**0.98**	0.97	**0.06**	0.94
c)	0.81	**0.76**	0.90	0.93	0.92	0.15	0.85

Considered configurations are given in Sec. V-2. Colored values indicate best achieved results. (acc_{Set}: Top-1 accuracy; acc_{CM}: Mean Class accuracy; FPR: False Positive Rate; TNR: True Negative Rate)

TABLE IV
RESULTS FOR GENERALIZATION AND DATASET COMBINATION EXPERIMENTS

Training	Test	acc_{Set}	acc_{CM}	prec.	recall	f_1	FPR	TNR
KITTI	KITTI	**0.88**	**0.86**	0.95	0.98	0.97	0.08	0.92
TUBS	KITTI	0.80	**0.51**	0.90	0.94	0.92	0.18	0.82
KITTI	TUBS	**0.53**	**0.61**	0.68	0.93	0.79	0.64	0.36
mixed	mixed	**0.90**	**0.89**	0.93	0.97	0.95	0.10	0.90

Blue-colored values indicate best achieved results, red-colored ones indicate degraded performance compared to using only TUBS datasets (cf. Sec. V). (acc_{Set}: Top-1 accuracy; acc_{CM}: Mean Class accuracy; FPR: False Positive Rate; TNR: True Negative Rate)

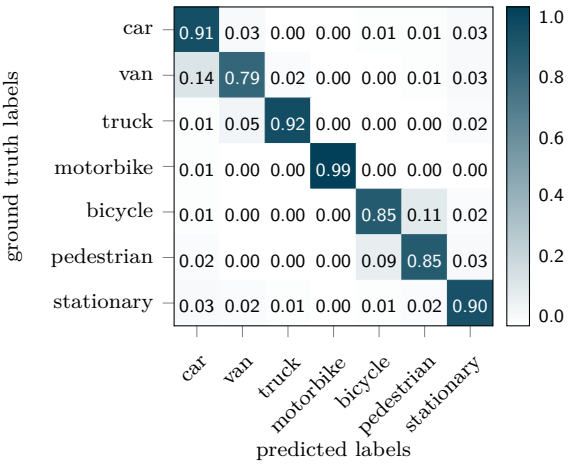

Fig. 7. Confusion matrix for the TUBS/KITTI mixed dataset setup

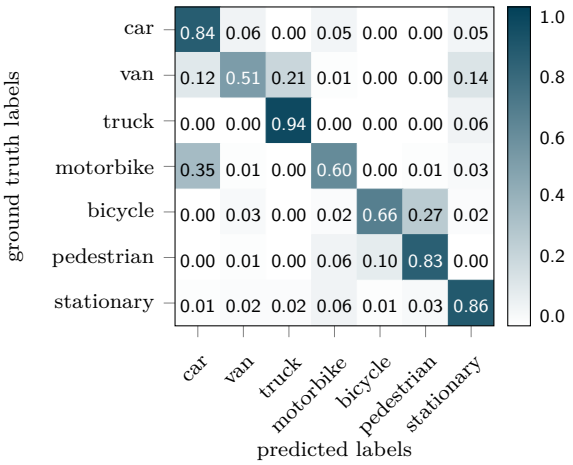

Fig. 8. Confusion matrix for results with automatically labeled data

regarding class and specific range. For all generalization experiments we used 25 000 iteration steps due to more available samples originating from the KITTI dataset.

VI. Results using automatically labeled data

To evaluate the performance of a network trained on automatically labeled data only, we trained a CNN using the best global setup for acc_{CM} found in Sec. V. The training set includes ≈1.5 mio. automatically labeled samples. We applied an online validation using the manually labeled validation set. We uniformly selected the same number of randomly chosen samples for each class to get an approximate value of acc_{CM} during training. For testing we used the same dataset as in Sec. V. Training was stopped after 20 epochs for runtime reasons and the best-performing model was selected.

A. Dealing with label noise

Automated labeling using the vehicle's tracking system is of course imperfect. Therefore, the dataset is subject to label noise. As opposed to the usage of manually data, training with automatically labeled data does not converge without further measures. Hence, all of the following methods were applied simultaneously in order to deal with label noise.

a) Surrogate cost function: The cost function given in Eq. 2 was additionally weighted by an object's existence likelihood, which is applied as an additional factor to w_k.

b) Label smoothing: The usage of label smoothing implies the assumption of a prior uniform distribution regarding a sample's classification probability. Hence, the CNN is not forced to perform peak formed predictions, especially for falsely labeled samples. Szegedy *et al.* [26] apply label smoothing to improve generalization. Applying label smoothing in the presence of label noise improves convergence significantly without much effort.

c) Noise layer training: Sukhbaatar *et al.* [27] add an additional layer between the softmax output and actual targets to deal with label noise. Using a special training procedure, this allows the overall network to learn the label noise distribution. This equals a mapping of correct labels to the possibly wrong labels in the dataset and enables the CNN to learn the correct labels out of noisy training data. The utilization of this method yielded promising results that will be further inquired in future work. However, due to a long training process we omitted that method in this paper's experiments.

d) Increasing model depth: We found that our simple CNN architecture given in Fig. 3 had not enough capacity to cope with noisy labels. Therefore, model depth has been increased to ten layers, inspired by the VGG architectures by Simonyan and Zisserman [9].

B. Results

Fig. 8 shows the results for using automatically labeled data for training and the TUBS test dataset for evaluation. Note that the results are almost equal to the training with manually labeled data (Fig. 6). This demonstrates the capability of a deeper architecture to model label noise when using label smoothing. This is an indicator showing that utilizing automatically labeled data for training purposes performs on the same level as relying on manually labeled data only. However, training with automatically labeled data needs a more complex architecture, more samples and, therefore, more time.

VII. Conclusion

In this paper, we presented a new laser scan dataset that fills the gap of detailed road user labels in the scanner's entire field of view.

We applied this dataset to a road user classification task using a 2.5D CNN. By optimizing data representation and normalization strategies, as well as by combining different datasets, we achieved an overall accuracy of 0.89. We achieved similar results using automatically labeled data. This demonstrates the possibility to renounce manual labeling, although the model depth had to be increased to cope with noisy labels, among other measures. Our approach extends to different data sources, as an evaluation on the well-known KITTI dataset has shown.

In future works we will improve our network architecture, e. g. by down-sampling close objects to use a smaller input size in order to reduce the network's parameters. Furthermore, we will experiment with deeper architectures and compare our results to 3D CNN approaches. The datasets and the labeling tool will be published by the end of this year. We encourage the publication of more manually edited laser scans. Therefore, we invite the users of our labeling tool to share their work by contacting us.

References

[1] J. Campbell, H. Ben Amor, M. H. Ang, and G. Fainekos, "Traffic light status detection using movement patterns of vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 283–288.

[2] A. Khosroshahi, E. Ohn-Bar, M. Mohan, and M. Trivedi, "Surround Vehicle Trajectory Analysis with Recurrent Neural Networks," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 2267–2272.

[3] T. Kim and J. Ghosh, "Robust detection of non-motorized road users using deep learning on optical and LIDAR data," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 271–276.

[4] M. Limmer, J. Forster, D. Baudach, F. Schüle, R. Schweiger, and H. P. A. Lensch, "Robust Deep-Learning-Based Road-Prediction for Augmented Reality Navigation Systems at Night," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 1888–1895.

[5] S. Martin, M. Trivedi, and K. Yuen, "On Looking at Faces in a Vehicle with Deep Networks," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 649–654.

[6] G. Zhong, Y.-H. Tsai, Y.-T. Chen, X. Mei, D. Prokhorov, M. James, and M.-H. Yang, "Learning to tell brake lights with convolutional features," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 1558–1563.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, U.S.A.: IEEE, 2014, pp. 1–9.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, U.S.A., 2015.

[10] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany: IEEE, 2015, pp. 681–687.

[11] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, U.S.A.: IEEE, 2015, pp. 1329–1335.

[12] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012, pp. 656–664.

[13] L. A. Alexandre, "3D Object Recognition Using Convolutional Neural Networks with Transfer Learning Between Input Channels," in *Intelligent Autonomous Systems 13*, ser. Advances in Intelligent Systems and Computing, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds., vol. 302, Cham: Springer International Publishing, 2016, pp. 889–898.

[14] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First Fully Autonomous Test Drives in Urban Traffic," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., U.S.A.: IEEE, 2011, pp. 919–924.

[15] J. Rieken, R. Matthaei, and M. Maurer, "Toward Perception-Driven Urban Environment Modeling for Automated Road Vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Gran Canaria, Spain: IEEE, 2015, pp. 731–738.

[16] J. Rieken and M. Maurer, "Sensor scan timing compensation in environment models for automated road vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil: IEEE, 2016, pp. 635–642.

[17] L.-C. Caron, Y. Song, D. Filliat, and A. Gepperth, "Neural network based 2D/3D fusion for robotic object recognition," in *European Symposium on artificial neural networks (ESANN)*, Bruges, Belgium: ESANN organization, 2014, pp. 127–132.

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

[19] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, U.S.A.: IEEE, 2016, pp. 6526–6534.

[20] A. Dewan, G. L. Oliveira, and W. Burgard, "Deep Semantic Classification for 3D LiDAR Data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada: IEEE, 2017, pp. 3544–3549.

[21] A. Zelener and I. Stamos, "CNN-Based Object Segmentation in Urban LIDAR with Missing Points," in *International Conference on 3D Vision (3DV)*, Stanford, CA, U.S.A.: IEEE, 2016, pp. 417–425.

[22] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud," in *Computing Research Repository (CoRR), arXiv preprint*, 2017.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, U.S.A.: IEEE, 2012, pp. 3354–3361.

[24] B. Li, "3D Fully Convolutional Network for Vehicle Detection in Point Cloud," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada: IEEE, 2017, pp. 1513–1518.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile: IEEE, 2015, pp. 1026–1034.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, U.S.A.: IEEE, 2015, pp. 2818–2826.

[27] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training Convolutional Networks with Noisy Labels," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, U.S.A., 2015.