Fast LIDAR-based Road Detection Using Convolutional Neural Networks

Luca Caltagirone[1], Samuel Scheidegger[2], Lennart Svensson[3], Mattias Wahde[4]
{luca.caltagirone, samsch, lennart.svensson, mattias.wahde}@chalmers.se

***Abstract*— In this work, a deep learning approach has been developed to carry out road detection using only LIDAR data. Starting from an unstructured point cloud, top-view images encoding several basic statistics such as mean height and density are generated. By considering a top-view representation, road detection is reduced to a single-scale problem that can be addressed with a simple and fast convolutional neural network (CNN). The CNN is specifically designed for the task of pixel-wise semantic segmentation by combining a large receptive field with high-resolution feature maps. The proposed system achieves state-of-the-art results on the KITTI road benchmark. It is currently the top-performing algorithm among the published methods in the overall category *urban road* and outperforms the second best LIDAR-only approach by 7.4 percentage points. Its fast inference makes it particularly suitable for real-time applications.**

I. INTRODUCTION

The estimation of free road surface (henceforth: road detection) is a crucial component for enabling fully autonomous driving [1]. Besides obstacle avoidance, road detection can also facilitate path planning and decision making, especially in those situations where lane markings are not visible (for example, because covered by snow or due to poor lightning conditions) or not present (for instance, in certain rural and urban roads).

The problem of road detection has been investigated for many years and a large variety of approaches can be found in the literature; see, for example, [1] for an in-depth survey of the field. Among the algorithms that perform best on the KITTI road benchmark data set [2], the large majority only work on monocular camera images and several make use of deep neural networks [3] (DNNs). For example, in [4] the author trains deep deconvolutional networks using a multi-patch approach, while in [5] a fully convolutional neural network is trained with automatically annotated images. Despite achieving state-of-the-art results, camera-based approaches are strongly affected by environmental illumination. As a consequence, their performance is expected to decrease considerably at night time or whenever presented with light conditions that deviate from those seen during training.

LIDARs, on the other hand, carry out sensing by using their own emitted light and therefore they are not sensitive to environmental illumination. Road detection systems that rely on this type of sensor can then, in principle, provide the same level of accuracy across the full spectrum of light conditions experienced in daily driving, and for this reason they are particularly suitable for achieving higher levels of driving automation. Several algorithms have been proposed that perform road detection exclusively in LIDAR point clouds or by fusing camera and LIDAR, see for example [6]–[9], but, to the best of our knowledge, none of them has used deep learning and their performance is consistently lower than the top-performing camera-based approaches.

In this paper, the problem of road detection is framed as a *pixel-wise semantic segmentation* task in point cloud top-view images using a fully convolutional neural network (FCN). The proposed system carries out road segmentation in real time, on GPU-accelerated hardware, and achieves state-of-the-art performance on the KITTI road benchmark.

The paper is organized as follows: In Section II, an overview and motivation of the proposed road detection system is presented and it is followed by a description of the procedure to transform an unstructured point cloud into top-view images in Section II-A. The CNN architecture is presented in Section II-B. The data set, data augmentation, and details about the training of the model are described in Section III. The results and a discussion are presented in Section IV and are followed by the conclusions in Section V.

II. POINT CLOUD TOP-VIEW ROAD DETECTION

The goal of this work is to perform road detection using only LIDAR data within a deep learning framework. Here, road detection is intended as the estimation of *all available* free surface for driving. Therefore, the differentiation of ego-lane versus oncoming or same-direction traffic lanes is not considered.

Starting from an unstructured point cloud, top-view images of the vehicle's surroundings are generated. Each image encodes one of several basic statistics such as, for example, mean height and mean reflectivity. A top-view representation is, in our opinion, more appropriate than a camera perspective representation given that both path planning and vehicle control are executed in this 2D world [2]. Furthermore, by using top-view images, classification is reduced to a simpler single-scale problem considering that patches of a given size cover equal surface area regardless of their position in the image. An analogous procedure for generating top-view images was also used by Chen *et al.* [10] but in the context of 3D object detection.

[1]Luca Caltagirone and [4]Mattias Wahde are with the Adaptive Systems Research Group, Applied Mechanics Department, Chalmers University of Technology, Gothenburg, Sweden. [2]Samuel Scheidegger and [3]Lennart Svensson are with the Signal and Systems Department, also at Chalmers University of Technology. [2]Samuel Scheidegger is also with Autoliv Research.

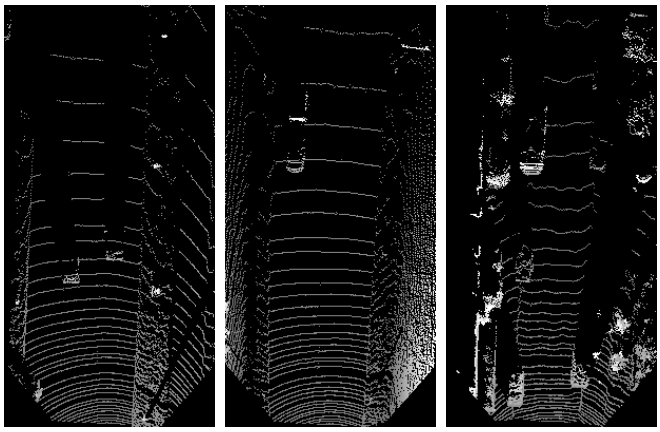

Fig. 1. Three examples of mean height images: Each pixel grayscale value encodes the mean height of its relative grid cell.

A CNN, specifically developed for semantic segmentation, is then trained to carry out road detection in the top-view images. The model is fully convolutional and can therefore process images of any size. An advantage of this design choice is that road detection can be carried out in regions of interest (ROIs) that can be dynamically changed and, in the case of rotating LIDARs, can even span a 360° view around the vehicle.

A. From point cloud to top-view images

An unstructured point cloud must be transformed into a suitable format before it can be used as input for a CNN. The first step of the procedure is to create a grid in the x-y plane of the LIDAR and to assign each element of the point cloud to one of its cells. The grid covers a region which is 20 meters wide, $y \in [-10, 10]$, and 40 meters long, $x \in [6, 46]$, as required for the evaluation of the KITTI road benchmark; its cells are squares of size 0.10×0.10 meters.

Some basic statistics are then computed for each grid cell: number of points; mean reflectivity; as well as mean, standard deviation, minimum, and maximum height. Finally, six images, one for each of the above statistics, are generated by viewing the grid cells as pixels. Figure 1 shows three examples of *mean height* images obtained with this procedure. Given the chosen cell size and grid range, these top-view images have a resolution of 200×400 pixels.

B. Model architecture

In recent years, deep learning models have achieved state-of-the-art results in several semantic segmentation benchmarks (e.g., PASCAL VOC [11], MSCOCO [12], etc.). The core idea many of those models share is that they start from a pretrained network for image classification (e.g., VGGNet [13]) which acts as a feature extractor, or encoder. Additional specialized layers, such as max unpooling and deconvolution, are then added to the network in order to upsample the feature maps back to the original input size. Some well-know deep networks that use this approach and that have inspired our model's architecture are Segnet [14], FCN-8s [15], and Dilation [16].

In this work, however, considering that point cloud and camera images are fundamentally different, it was deemed more appropriate to train the model from scratch, using only KITTI training data (see Section III-A), instead of starting from a pretrained encoder. This decision provided freedom to implement a network architecture that is specifically designed for semantic segmentation and that is tailored to the problem at hand. Particularly, the CNN was designed to have a large receptive field and to process high-resolution feature maps, two aspects that have been shown to improve segmentation accuracy [17]. The model's architecture is shown in Fig. 2 and consists of the following components:

1) A six-channel input layer, one channel for each of the point cloud statistics described in Section II-A.
2) An encoder whose main purpose is to subsample the feature maps, thus reducing the model memory requirements. Subsampling is carried out by using a max pooling layer with a 2×2 window and stride 2.
3) A context module that aggregates multi-scale contextual information by using dilated convolutions [16]. More details are provide in Section II-C.
4) A decoder that upsamples the feature maps back to the input size by using a max-unpooling layer [14] followed by two convolutional layers.
5) An output layer that returns a road confidence map, that is, an image where each pixel's value represents the probability that its corresponding grid cell in the LIDAR x-y plane (see Section II-A) belongs to the road.

C. Context module

An efficient strategy to expand the receptive field of a CNN while keeping the number of model parameters and layers small is to employ the dilated convolution operator which supports an exponential expansion of the receptive field without losing resolution (i.e., the feature maps do not decrease in size) or coverage [16]. Restricting the number of layers is important in order to reduce the model memory requirements, especially when working with high-resolution feature maps.

Table I shows the implemented context module architecture; as can be seen, the receptive field of the last dilated convolution layer is larger than the input feature maps, which have a size of 100×200 pixels. This allows the CNN to access a large context window for inferring whether a pixel belongs to the road or not, which is particularly important considering the sparsity of point cloud top-view images (see Fig. 1).

III. DATA SET AND SETUP

A. The KITTI data set

The KITTI road benchmark data set [2] consists of 289 training images and 290 test images taken over several days in various locations: city, rural, and highway. Ground truth annotations are represented in the camera perspective space and are only available for the training set. LIDAR point clouds are also provided as an extension to the data set.

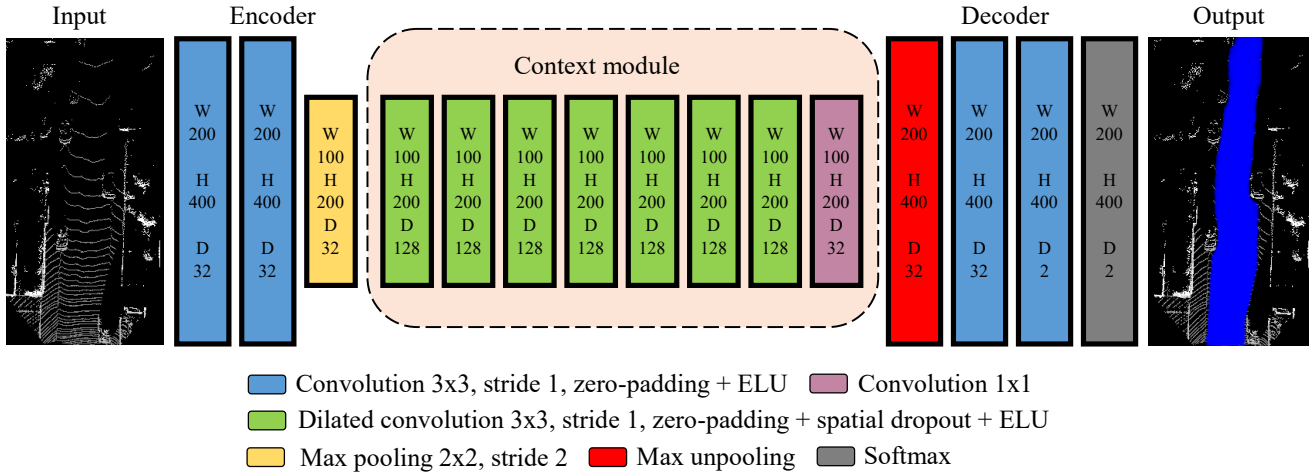

Fig. 2. A schematic illustration of the proposed model. The input consists of 6 stacked images, one for each of the point cloud statistics described in Section II-A. The output is a road confidence map: The value of each pixel represents the probability that the corresponding grid cell in the LIDAR x-y plane belongs to the road. W represents the width, H denotes the height, and D is the number of feature maps. The model uses the exponential linear unit (ELU) activation function [18].

TABLE I

CONTEXT MODULE ARCHITECTURE. BY USING AN EXPONENTIALLY GROWING DILATION, THE RECEPTIVE FIELD ALSO GROWS EXPONENTIALLY WITHOUT LOSING COVERAGE. THE FEATURE MAPS ARE ZERO-PADDED SO THERE IS NO LOSS OF RESOLUTION. THE RECEPTIVE FIELD GROWS AT DIFFERENT RATES IN WIDTH AND HEIGHT, MATCHING THE INPUT IMAGES' ASPECT RATIO (1:2).

Layer	1	2	3	4	5	6	7	8
Filter size	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation (width, height)	$(1, 1)$	$(1, 2)$	$(2, 4)$	$(4, 8)$	$(8, 16)$	$(16, 32)$	$(32, 64)$	-
Receptive field	3×3	5×7	9×15	17×31	33×63	65×127	129×255	129×255
# Feature maps	128	128	128	128	128	128	128	32
Non-linearity	ELU	ELU	ELU	ELU	ELU	ELU	ELU	-

The examples are divided into three approximately equally sized (see Table II) categories: urban unmarked (uu), urban marked (um), and urban multiple marked lanes (umm). In this work, 30 examples have been assigned to the validation set, 10 for each category.

The training set is used for computing the objective function and adjusting the CNN weights, while the validation set is used to decide when to stop training. Moreover, the validation set is also used for selecting the CNN hyper-parameters (such as, for example, number of layers, filters' size, and learning rate) by choosing the model (from a large set of runs) with the smallest validation error. The test set is only used for evaluating the model performance on unseen data, that is, its generalization error.

TABLE II

KITTI ROAD DATASET: SIZE AND NUMBER OF IMAGES FOR EACH CATEGORY AND SPLIT.

Category	Train	Validation	Test	Size [px]
urban marked	85	10	100	200×400
urban multiple marked	86	10	94	200×400
urban unmarked	88	10	100	200×400

B. Data augmentation

Given that the models were trained using only the KITTI data set, some simple data augmentation was necessary in order to avoid overfitting and to improve model generalization. For this purpose, each training example was rotated about the LIDAR z-axis for angles in the range $[-30°, 30°]$ using steps of three degrees. After rotation, each example was also mirrored about the x-axis. In this way, the data set size was increased by a factor of 42.

C. Inverse perspective mapping vs. point cloud projection

As previously mentioned, ground truth annotations provided with the KITTI data set are represented in the camera perspective. However, given that the proposed system works with top-views of the road, the annotations must be transformed to that space for training. A possible approach to accomplish this is to use a technique known as inverse perspective mapping (IPM). Unfortunately, IPM makes the assumptions of flat and obstacle-free roads which are rarely satisfied in the real world. As a consequence, it often produces images showing inaccurate distances and road geometries. An example of this problem is illustrated in Fig. 3.

An alternative approach is to project the point cloud into the corresponding camera-view annotation in order to determine which of its points belong to the road and then use a procedure similar to the one described in Section II-A but considering the class instead of the height and reflectivity statistics. To increase the density of points and obtain a dense annotation, the point cloud is interpolated linearly within

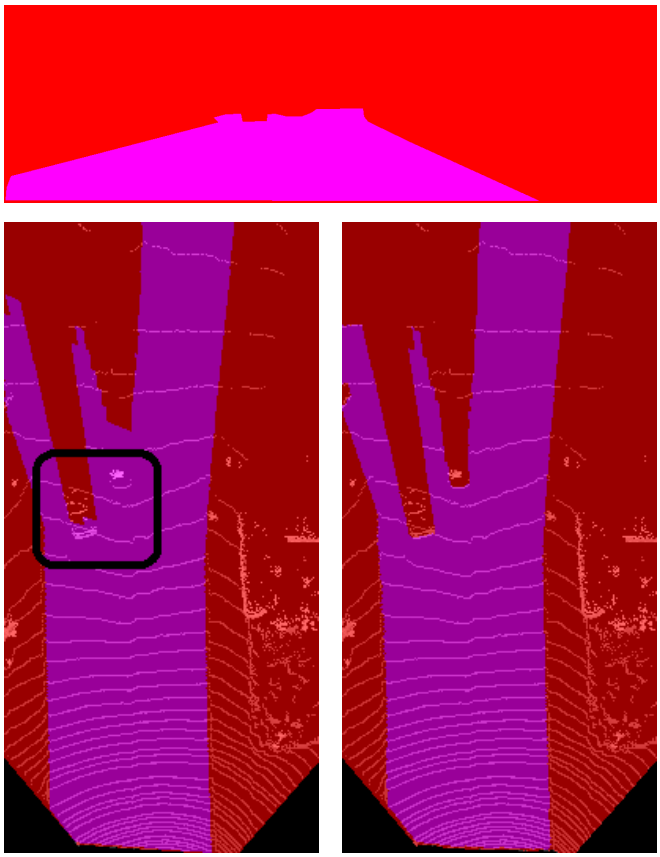

Fig. 3. Top panel: Annotation in the camera perspective space. Violet and red represent road and not-road, respectively. Left panel: An example of mismatch between a point cloud top-view image and the corresponding IPM-generated annotation . The black rectangle highlights a case where the rear of a vehicle and a curb fall in the road-labeled region. Right panel: Corresponding top-view annotation obtained by projecting the interpolated point cloud onto the perspective annotation image. Here, the curb and the rear of the vehicle are both correctly marked as not-road.

narrow circular sectors before carrying out the projection. The right panel of Fig. 3 shows an example of top-view annotation obtained by using this procedure.

D. Training

The CNN was trained using the Adam optimization algorithm [19] with an initial learning rate of 0.01, and using cross-entropy loss as the objective function. The cross-entropy loss is defined as

$$L = -\frac{1}{N \times W \times H} \sum_{i=1}^{N} \sum_{m=1}^{W} \sum_{n=1}^{H} \log p_{m,n}^{i} \qquad (1)$$

where W and H represent, respectively, the width and height of the softmax layer's output, and N is the batch size which in this work was set to 4. The variable p is the probability predicted by the CNN for the correct class. The learning rate was decayed by a factor 2 whenever there was no improvement of performance within the last epoch. For regularization, spatial dropout layers ($p_d = 0.25$) [20] were added after each dilated convolution layer. The model was implemented using the Torch7 framework and was trained on an NVIDIA GTX980Ti GPU with 6GB of memory.

IV. EXPERIMENTS

A. KITTI road benchmark

The proposed road detection system was evaluated on the KITTI road benchmark test set. As mentioned in Section III-A, this set consists of three categories: urban unmarked (uu), urban marked (um), and urban multiple marked lanes (umm). In addition, a category called *urban road* is computed, which provides an overall score for these three categories combined. The metrics used for evaluation are precision (PRE), recall (REC), false positive rate (FPR), false negative rate (FNR), average precision (AP), and maximum F1-measure (MaxF) which is defined as follows:

$$\text{MaxF} = \max_{\tau} \; 2 \times \frac{\text{PRE}(\tau) \times \text{REC}(\tau)}{\text{PRE}(\tau) + \text{REC}(\tau)}, \qquad (2)$$

where τ is the classification threshold.

As shown in Table III, the proposed road detection system achieved state-of-the-art performance on the KITTI road benchmark; in fact, it is currently the top-performing algorithm among the published methods in the overall category *urban road* and it outperformed by 7.4 percentage points the second best LIDAR-only system. Furthermore, its inference time is significantly smaller than most other approaches which makes it suitable for real time deployment on GPU accelerated hardware. Figure 4 shows some road segmentations generated by the proposed model on examples from the test set. As is evident from the figure, the boundary between regions that have a high probability of being part of the road and those that do not is very sharp, making the resulting road region almost uniformly blue. The results on individual categories and additional evaluation metrics can be found at the KITTI road benchmark web page[1]. Some examples of road segmentations projected onto the camera images are illustrated in Fig. 5. Several road detection videos can be found at `http://goo.gl/efLoHz`.

TABLE III
KITTI ROAD BENCHMARK RESULTS (IN %) ON URBAN ROAD CATEGORY. ONLY RESULTS OF PUBLISHED METHODS ARE REPORTED.

Model	MaxF	AP	PRE	REC	Time (ms)
LoDNN (our)	**94.07**	**92.03**	92.81	**95.37**	**18**
Up-Conv-Poly [21]	93.83	90.47	94.00	93.67	80
DDN [4]	93.43	89.67	**95.09**	91.82	2000
FTP [5]	91.61	90.96	91.04	92.20	280
FCN-LC [22]	90.79	85.83	90.87	90.72	30
HIM [23]	90.64	81.42	91.62	89.68	7000
NNP [24]	89.68	86.50	89.67	89.68	5000
RES3D-Velo [9]	86.58	78.34	82.63	90.92	360

At times, false positive detections were observed in situations where the boundary between road and sidewalk was not sharp such as, for example, when the sidewalk merged with the road at pedestrian crossings, or, as illustrated in Fig. 6, when the difference in height between road and sidewalk was very small or negligible. Complex road scenes, such as intersections, also resulted in unclear segmentations in

[1] `http://www.cvlibs.net/datasets/kitti/eval_road.php`

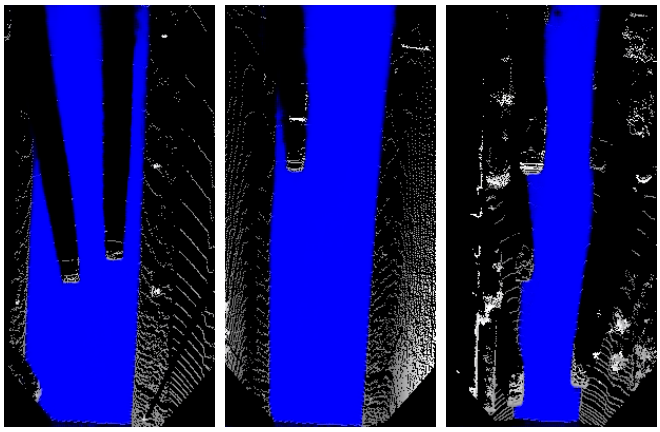

Fig. 4. Road detections generated by our model corresponding to the mean height images shown in Fig. 1. Higher blue intensity pixels correspond to higher probability road regions.

Fig. 5. Examples of road detection in images from the test set. Green denotes true positive; red and blue correspond to false negative and false positive, respectively.

some cases. These problems may be reduced by extending the training set to include more examples of such situations, using annotations explicitly made for road detection in point cloud top-views, and considering additional features for generating top-view images.

B. Regions of interest study

LIDAR-acquired point clouds are sparse and have densities that decrease with distance from the sensor. It is therefore of relevance to evaluate how performance is affected when considering smaller ROIs with higher density of points. According to the results shown in Table IV, the model performs best when considering regions up to 31 meters away. After that, performance degrades steadily and reaches its lowest value at the maximum considered distance of 46 meters. In order to deal with low point density in regions farther away from the LIDAR, a possible solution would be to accumulate points over successive scans. However, this is not trivial because of the presence of dynamic objects in the surroundings, such as other vehicles, as well as uncertainties introduced when estimating the ego-vehicle motion.

TABLE IV

ROI SIZE STUDY. THE RESULTS PERTAIN TO THE VALIDATION SET. THE y-RANGE IS [-10, 10] METERS IN ALL CASES. THE x-RANGE LOWER BOUND IS 6 METERS IN ALL CASES.

x-upper bound [m]	MaxF	PRE	REC	FPR	FNR
46	95.58	94.15	97.05	3.33	2.86
41	95.90	94.36	97.50	3.33	2.42
36	96.13	94.54	97.78	3.37	2.15
31	96.34	94.75	97.99	3.50	1.95
26	96.37	94.78	98.02	3.80	1.93
21	96.36	94.67	98.11	4.19	1.92

C. Point-cloud vs. IPM annotations

As explained in Section III-C, top-view annotations obtained by using IPM often do not match properly with their corresponding point cloud top-view images. Given that the evaluation for the KITTI road benchmark is carried out using IPM annotations, our system is thus penalized compared with other approaches. In order to estimate the loss of performance, a comparison has been made on the validation set using both mapping strategies. As shown in Table V, by using the more accurate annotations obtained by projecting the interpolated point cloud, our model performance increases significantly in all the considered metrics. Precision, in particular, sees an increase of 1.2 percentage points compared with the score obtained using the IPM annotations. It is therefore likely that the proposed system would achieve higher performance also on the test set if it were evaluated using more accurate top-view annotations.

TABLE V

COMPARISON IPM VS. POINT CLOUD PROJECTION (PCP) FOR GENERATING TOP-VIEW ANNOTATIONS.

Split	Mapping	MaxF	PRE	REC	FPR	FNR
Validation	PCP	95.58	94.15	97.05	3.33	2.86
Validation	IPM	94.51	92.92	96.16	3.94	3.66
Test	IPM	94.07	92.81	95.37	4.07	4.63

D. Occupancy images

The statistics used for generating the point cloud top-view images were selected because they are simple and fast to compute, and only require information contained in the individual grid cells. However, it is possible to consider additional and more complex features that take into account neighboring cells or the spatial distribution of points within individual cells (e.g., principal components) and which may provide higher segmentation accuracy. Furthermore, it is far from obvious that the selected statistics and their combination is optimal for road detection. In future work, these issues may be explored more in-depth.

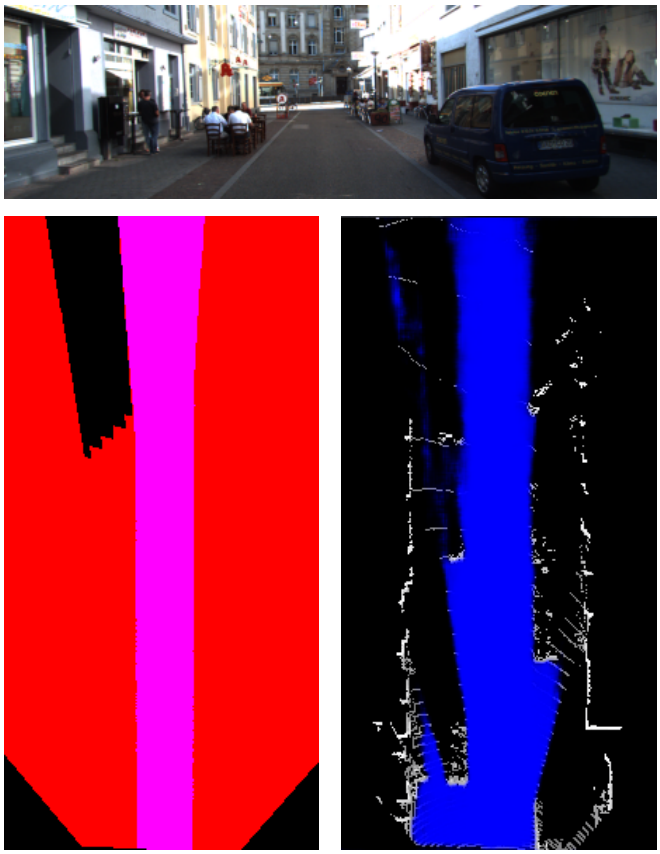

Fig. 6. Top panel: Perspective camera image. Left bottom panel: Ground truth annotation. Right bottom panel: Road segmentation generated by the proposed system. The ground truth annotation labels as road only the central lane whereas the CNN's segmentation extends further to the sides which are also drivable, as illustrated by the presence of another vehicle, but that are indeed sidewalks.

An interesting preliminary result is that the CNN is able to achieve high performance: MaxF = 95.32%, PRE = 94.15%, REC = 96.52%, using as input only occupancy images (i.e., white pixel if there is at least one detection, black otherwise). This indicates that the 2D distribution of points by itself, as seen from a top-view perspective, already contains strong discriminative information for road detection.

V. CONCLUSION

In this work a CNN has been developed to perform road detection in point cloud top-view images. The network achieves state-of-the-art performance on the KITTI road benchmark, while only making use of LIDAR data, and therefore it can provide high accuracy road segmentations in any lighting conditions. Furthermore, it works in real time on GPU-accelerated hardware. Both these features make it particularly suitable for being integrated into high-level driving automation systems.

ACKNOWLEDGMENT

The authors gratefully acknowledge financial support from Vinnova/FFI. This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

REFERENCES

[1] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[2] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] R. Mohan, "Deep deconvolutional networks for scene parsing," *arXiv preprint arXiv:1411.4101*, 2014.

[5] L. Ankit, K. Mehmet, S. Luis, and M. Hebert, "Map-supervised road detection," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2016.

[6] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "Crf based road detection with multi-sensor fusion," in *Intelligent Vehicles Symposium (IV)*, 2015.

[7] X. Hu, F. S. A. Rodriguez, and A. Gepperth, "A multi-modal system for road detection and segmentation," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1365–1370.

[8] R. Fernandes, C. Premebida, P. Peixoto, D. Wolf, and U. Nunes, "Road detection using high resolution lidar," in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Oct 2014, pp. 1–6.

[9] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 687–692.

[10] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *arXiv preprint arXiv:1611.07759*, 2016.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[12] T. 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, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[16] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[17] Z. Wu, C. Shen, and A. van den Hengel, "High-performance semantic segmentation using very deep fully convolutional networks," *CoRR*, vol. abs/1604.04339, 2016. [Online]. Available: http://arxiv.org/abs/1604.04339

[18] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[21] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep methods for monocular road segmentation." in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, 2016.

[22] C. C. T. Mendes, V. Frmont, and D. F. Wolf, "Exploiting fully convolutional neural networks for fast road detection," in *IEEE Conference on Robotics and Automation (ICRA)*, May 2016.

[23] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *European Conference on Computer Vision (ECCV)*, 2010.

[24] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *NIPS*, 2015.