Improvement of projection-based LiDAR data segmentation algorithms using object-contextual representations

M Stokolesov* and D Yudin

1Faculty of Information Technology and Applied Mathematics, Moscow Aviation Institute (National Research University), 4 Volokolamskoye Highway, 125993, Moscow, Russian Federation
2Center for Cognitive Modeling, Moscow Institute of Physics and Technology, 9 Institutsky Lane, 141700, Dolgoprudny, Moscow Region, Russian Federation

*E-mail: MSStokolesov@mai.education

Abstract. Nowadays self-driving cars and such unmanned aerial vehicles as drones are one of the most actively developed technologies, where machine learning algorithms play an irreplaceable role especially in the perception problem, which is the context of this research. To be applicable in self-driving cars and especially drones such algorithms should not only have good output quality, but also be real-time. For this reason, in the case of LiDAR data segmentation problem we pay special attention to algorithms that are based on point cloud projections because of their speed superiority over other heavy algorithms that process input point cloud directly. The main drawback of projection-based algorithms is their lower segmentation accuracy, so in this paper we show that it can be improved by integrating contextual representation module inside segmentation algorithm architecture. In our work we consider SalsaNext as a segmentation algorithm and OCR as a context representation module because these methods are among the highest in the corresponding datasets’ leaderboards. We provide results from quantitative evaluation on the Semantic-KITTI dataset, which demonstrate that the proposed SalsaNext modification gives 6.2% mean intersection over union metric improvement with no speed reduction.

1. Introduction
Self-driving cars and such unmanned aerial vehicles as drones are currently among the most popular and rapidly developing technologies, and there is a very important problem of object recognition around an autonomous vehicle. The most common sensor for collecting such information is a stereo camera, but there are many examples of situations when a single camera may not be enough, such as fog, heavy rain, snowfall, and even at night time. Therefore, in order to correct these flaws, the car or drone is equipped with an additional perception device, namely a LiDAR.

For this reason, LiDAR point cloud semantic segmentation is one of the most important steps in the scene understanding problem that provides rich information about the autonomous vehicle’s surrounding. This information is used for solving such self-driving problems as SLAM [1] and motion planning [2].

To be applicable in self-driving cars and drones, segmentation algorithms should have high accuracy with low latency at the same time. However, we cannot give arbitrary computing power to self-driving cars and especially drones, since at present this would not only significantly increase the cost of such systems, but it would also make it less useful in practical terms, since powerful hardware requires a lot...
of space. And this is the reason it’s important to develop algorithms that can efficiently applied with limited computing power.

Currently, there are two main deep learning approaches for LiDAR data segmentation (according to [3,4]): point-wise and projection-based neural networks. First approach operates directly on the raw 3D points or it’s volumetric representation, whereas the second uses various point cloud flat projections.

And we pay special attention to LiDAR data segmentation algorithms that are based on point cloud projections (see figure 1 on the bottom) because of their superiority over other heavy algorithms that process input point clouds directly in terms of low latency. This transition from a 3D point cloud to a 2D image makes it possible to actively use the advantages of deep convolutional neural networks, which have proven themselves well for object detection [5], semantic [6] and instance segmentation [7].

The main drawback of projection-based algorithms is their slightly lower segmentation accuracy [3], so there were several attempts to deal with it in [8], and in this paper we show that it can be improved by integrating object-contextual representation module inside segmentation algorithm. In our work we consider SalsaNext [9] as a target segmentation algorithm and OCR [10] as a context representation module to integrate because they are among the highest in the corresponding datasets’ leaderboards [11,12]. We address segmentation problem in the case of self-driving cars, but all results presented here can be easily applied for such unmanned aerial vehicles as drones, equipped with LiDAR.

This paper discusses the problem of multi-beam 360-degrees LiDAR point cloud segmentation, which are widely used in robotics and unmanned vehicles. Such LiDARs usually generate 2048 points for each beam (horizontal resolution). The total number of beams (vertical resolution) can be 128, 64, 32, 16, etc. depending on the sensor model. Each point contains information about x, y, z, depth, and intensity of diffuse reflection on surfaces (remission).

It is required to investigate the possibilities of improving the quality and ensuring a high segmentation speed, which allows each point from the cloud to be assigned one of 19 categories: car, bicycle, motorcycle, truck, other-vehicle, person, bicyclist, motorcyclist, road, parking, sidewalk, other-ground, building, fence, vegetation, trunk, terrain, pole, traffic-sign. These categories were selected because they are included in the very popular large-scale Semantic KITTI dataset [11].

To evaluate the results of our model we use the intersection-over-union (IoU) metric (or Jaccard Index) over all classes that is given by formula (1) [9]:

\[
miou = \frac{1}{C} \sum_{i=1}^{C} \frac{|P_i \cap G_i|}{|P_i| + |G_i| - |P_i \cap G_i|}
\]

where \(P_i\) is the set of points with a class prediction \(i\), \(G_i\) is the set of points with ground truth class \(i\) and \(|\cdot|\) means the cardinality of the set.

So the aim of this paper is to integrate OCR [10] module inside SalsaNext [9] architecture and obtain quantitative experimental results in terms of speed and mean IoU metric improvement.

2. Methodology
In our work we consider SalsaNext [9] as a target projection-based segmentation algorithm because of its superiority over other state-of-the-art algorithms in terms of speed (see section 3). SalsaNext is a LiDAR point cloud semantic segmentation neural network, which is an improvement of the RangeNet++ [13] approach, whose main idea is to use a spherical projection as a lightweight representation of the original point cloud, that significantly reduces inference time. This projection has the same properties as a regular image, so this allows to apply standard methods such as convolution and pooling without any changes. SalsaNext has an encoder-decoder architecture, where the encoder part consists of a series of ResNet [14] blocks with dropout and pooling. The decoder part makes upsampling via transposed convolution and fuses features extracted in the residual blocks. Each convolution layer in the SalsaNext architecture uses leaky-ReLU activation function and batch normalization for internal covariate shift treatment. To deal with class imbalance problem SalsaNext uses weighted softmax cross-entropy loss. The input of SalsaNext is rasterized image of spherical projection (formula (2) [9]) of the full LiDAR scan, where each image channel stores position, depth, and intensity values in the panoramic 360° view format (figure 1).

The output of SalsaNext is the point-wise classification scores, which are mapped
back from the image segmentation to the initial point cloud and refined by a special version of kNN algorithm for getting result segmentation on the final stage.

\[
\begin{align*}
(u, v) &= \left( \frac{\frac{1}{2} \left[ 1 - \arctan(y, x) \pi^{-1} \right] w}{1 - \arcsin (z, r^{-1}) + f_{\text{down}} f^{-1} h} \right), \\
\end{align*}
\]

where \((u, v)\) denotes image pixel coordinates of the corresponding LiDAR point \((x, y, z)\), \(h\) and \(w\) denote height and width of the projected image, \(r\) represents the range of each point as \(r = \sqrt{x^2 + y^2 + z^2}\), \(f\) defines the sensor vertical field of view as \(f = |f_{\text{down}}| + |f_{\text{up}}|\), and \(f_{\text{down}}\) with \(f_{\text{up}}\) denote LiDAR’s lower field of view and upper field of view respectively.

**Figure 1.** Example of spherical projection image (bottom part) constructed from source point cloud (middle part) inside SalsaNext [9] framework (Semantic KITTI dataset [11] cloud).

As shown in the network architecture diagram (figure 2), at the very beginning there is a so-called context module, the purpose of which is to aggregate the context information in different image regions. It’s performed by fusing a larger receptive field with a smaller one by adding \(3 \times 3\) and \(1 \times 1\) kernels right at the beginning of the network. This helps to capture the global context alongside with more detailed spatial information. But the drawback of such a module is that it does not take into account the feature information that accumulates in the network during the input image is passed through its layers. And because SalsaNext operates with a flat projection image, for solving this problem we can apply an OCR [10] approach that was designed to take into account exactly such features. We prefer to use OCR as one of the most powerful [12] context aggregation methods at the moment of writing this paper.

**Figure 2.** Proposed modification of SalsaNext [9] architecture with two possible positions for OCR [10] module integration and excluded intensity channel from the input image shape.
More precisely, OCR is intended to be used for refining coarse segmentation that can be extracted from the last layers of the backbone network. This allows us to take into account the context of each image pixel and produce more accurate segmentation. The key idea of OCR approach is to use object-contextual representations, characterizing a pixel by exploiting the representation of the corresponding object class. First, object regions are learned under the supervision of the ground-truth segmentation. Second, the object region representation is computed by aggregating the representations of the pixels lying in the object region. Last, the relation between each pixel and each object region is computed, and the representation of each pixel is augmented with the object-contextual representation which is a weighted aggregation of all the object region representations. OCR module architecture is presented on figure 3.

Figure 3. OCR [10] module (pink dashed box) integrated inside SalsaNext [9] architecture (blue dashed boxes). Red dashed box: form the soft object regions; Green dashed box: estimate the object region representations; Orange dashed box: compute the object contextual representations and the augmented representations.

3. Results and discussion
At the first, we have made a comparison in terms of speed and mean IoU metric between the top 3 accuracy-efficient algorithms from the leaderboard of the large-scale Semantic KITTI dataset [11], contained over 43K annotated full 3D LiDAR scans, that is currently the standard dataset for evaluating the quality of outdoor scene segmentation models. Evaluation on the KITTI 08 sequence (validation split) showed that SalsaNext network [9], which is a projection-based algorithm, is at least 2x faster than Cylinder3D [15] with SPVNAS [16] and has 1.13x lower mean IoU (table 1), that corresponds to metrics from Semantic KITTI leaderboard. Therefore we took SalsaNext as the target network for our research.

Table 1. Speed and accuracy comparison between top 3 point cloud segmentation algorithms on Semantic KITTI dataset [11]. Weights for tested models were provided by their authors. The evaluation was performed on the GPU Nvidia Tesla V100 32GB.

| Metric               | Cylinder3D | SPVNAS | SalsaNext |
|----------------------|------------|--------|-----------|
| mean inference speed (single cloud, fps) | 8.65       | 7.38   | 17.5      |
| mean IoU             | 0.669      | 0.646  | 0.590     |

In our case we use SalsaNext network as backbone, which provides coarse segmentation for OCR module. But if we try to apply the module in a standard way, i.e. at the end of the network, then we will face difficulties because this approach requires a lot of device memory that network runs on. For example, 32 GB memory space of GPU Nvidia Tesla V100 isn't enough to train such architecture in a reasonable time. Thus we made an attempt to apply the OCR module in the centre network layers, where we have an image with much smaller dimensions than on the last layer after a cascade of convolutions: [64 × 2] × 1024 and [256 × 8] × 256 vs [2048 × 64] × 32 correspondingly in case of SalsaNext (figure 2). Additionally, we modified the SalsaNext input image shape to describe each point with 4
parameters \((x, y, z, \text{depth})\) instead of 5 \((x, y, z, \text{depth, intensity})\) as in original paper. This was done to increase the generalization ability of SalsaNext by preventing overfitting to intensity values range of specific LiDAR, which allows us to test trained networks on datasets collected by LiDARs with different parameters without large accuracy loss (figure 4).

Thus, in this paper we investigate three SalsaNext modifications (figure 2):

- Default SalsaNext (with 4-channel input image);
- SalsaNext with integrated OCR module at first position;
- SalsaNext with integrated OCR module at second position.

Figure 4. SalsaNext [9] segmentation comparison on the same point cloud scene that was collected with Velodyne HDL-32 LiDAR: left – network was trained on Semantic KITTI dataset [11] (Velodyne HDL-64 LiDAR) with intensity channel, right – trained on the same dataset without intensity channel.

We trained all three SalsaNext modifications on the same Semantic KITTI dataset for 80 epochs with learning rate equal to 0.05 and other default parameters derived from the original paper [9]. We follow exactly the same protocol as source SalsaNext paper [9] and divide the dataset into training, validation, and test splits. For training we used sequences from 00 to 10, which contain over 21K scans. For validation we used sequence 08 (over 4K scans).

Best models’ results of evaluation on the 08 sequence are presented in the table 2. The computational experiments were performed on the workstation with GPU Nvidia Tesla V100 32GB.

As we see from evaluation results presented in the table 2, SalsaNext + OCR (I) and SalsaNext + OCR (II) modifications outperforms default SalsaNext architecture for 2.5% and 6.2% correspondingly by mean IoU metric, which confirms the hypothesis that the closer to the end of the network OCR module is located, the greater quality gain it gives. At the same time, this module doesn’t reduce the model speed. Someone can think that the increase of 6.2% mean IoU on validation set is too small, but such a result is the property of this module, which, moreover, has an easily calculated upper limit of a possible quality gain, for more details see the original paper [10]. For this reason, this experiment can be considered as successful.

The proposed improvement can also be useful when implementing the considered point cloud segmentation model for the on-board computer vision system of robotic platforms or such unmanned aerial vehicles as drones, where high recognition quality with low delays in the algorithm execution play a key role.

4. Conclusion

In this paper we presented an idea of the object-contextual representations [10] module integration inside the projection-based point cloud segmentation algorithm based on SalsaNext [9] network architecture. We obtained results that show us such modification gives expected mean IoU improvement that has tendency to increase with closer to the end of the network OCR module is placed, which allows us to expect more accuracy gain in case of using OCR in traditional way i.e. with the last network layers. So this defines our further research direction, specifically, to find a way to use the OCR module most efficiently with limited hardware resources. Also, it should be noted that all results presented in this paper can be applied not only for self-driving cars but also for such unmanned aerial vehicles as drones, equipped with LiDAR.
Table 2. Best trained models’ performance comparison on the 08 sequence (validation split) of the Semantic KITTI dataset, where we see that SalsaNext + OCR (I) and SalsaNext + OCR (II) modifications outperforms default SalsaNext architecture for 2.5% and 6.2% correspondingly by mean IoU metric.

| Metric          | SalsaNext | SalsaNext + OCR (I) | SalsaNext + OCR (II) |
|-----------------|-----------|---------------------|----------------------|
| mean inference speed (single cloud, fps) | 17.5 | 17.5 | 17.5 |
| mean IoU        | 0.514 | 0.527 (+2.5%) | 0.546 (+6.2%) |
| car             | 0.928 | 0.927 | 0.928 |
| bicycle         | 0.078 | 0.157 | 0.182 |
| motorcycle      | 0.369 | 0.333 | 0.453 |
| truck           | 0.655 | 0.574 | 0.700 |
| other-vehicle   | 0.358 | 0.232 | 0.237 |
| person          | 0.521 | 0.599 | 0.593 |
| bicyclist       | 0.684 | 0.777 | 0.776 |
| motorcyclist    | 0.000 | 0.000 | 0.000 |
| road            | 0.904 | 0.917 | 0.922 |
| parking         | 0.355 | 0.293 | 0.390 |
| sidewalk        | 0.756 | 0.790 | 0.780 |
| other-ground    | 0.005 | 0.002 | 0.001 |
| building        | 0.822 | 0.850 | 0.844 |
| fence           | 0.467 | 0.498 | 0.509 |
| vegetation      | 0.799 | 0.822 | 0.827 |
| trunk           | 0.541 | 0.617 | 0.593 |
| terrain         | 0.696 | 0.695 | 0.715 |
| pole            | 0.444 | 0.543 | 0.525 |
| traffic-sign    | 0.385 | 0.382 | 0.398 |

Acknowledgement
This work was supported in part of theoretical investigation and methodology (sections 1, 2, 4) by the Government of the Russian Federation under Agreement No. 075-02-2019-967 and in part of experimental evaluation (section 3) by the Integrant LLC.

References
[1] Huang B, Zhao J, Liu J 2019 A Survey of Simultaneous Localization and Mapping, available at: https://arxiv.org/pdf/1909.05214.pdf
[2] González D, Pérez J, Milanés V and Nashashibi F 2015 A review of motion planning techniques for automated vehicles. IEEE Trans. on Intelligent Transportation Systems 1135 DOI: 10.1109/TITS.2015.2498841
[3] Guo Y, Wang H, Hu Q, Liu H, Liu L and Bennamoun M 2020 Deep learning for 3d point clouds: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 1 DOI: 10.1109/TPAMI.2020.3005434
[4] Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N and Terzopoulos D 2021 Image segmentation using deep learning: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 33596172 DOI: 10.1109/TPAMI.2021.3059968

[5] Belkin I, Tkachenko S and Yudin D 2019 Traffic sign recognition on video sequence using deep neural networks and matching algorithm. Int. Conf. on Artificial Intelligence: Applications and Innovations (Vrdnik Banja, Serbia) pp 349-354 DOI: 10.1109/IC-AIAI48757.2019.00013

[6] Yudin D, Adeshkin V, Dolzhenko A, Polyakov A and Naumov A 2020 Roof defect segmentation on aerial images using neural networks. In Int. Conf. on Neuroinformatics vol. 925 (Moscow, Russia) pp 175-183 DOI: 10.1007/978-3-030-60577-3_20

[7] Staroverov A, Yudin D, Belkin I, Adeshkin V, Solomentsev Y and Panov A 2020 Real-time object navigation with deep neural networks and hierarchical reinforcement learning. IEEE Access 8 195608 DOI: 10.1109/ACCESS.2020.3034524

[8] Gan L, Zhang R, Grizzle J, Eustice R and Ghaffari M 2020 Bayesian spatial kernel smoothing for scalable dense semantic mapping. IEEE Robotics and Automation Letters 5(2) 790 DOI: 10.1109/LRA.2020.2965390

[9] Cortinhal T, Tzelepis G, Aksoy E 2020 SalsaNext: Fast Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving , available at: https://arxiv.org/pdf/2003.03653v1.pdf

[10] Yuan Y, Chen X, Wang J 2019 Object-contextual representations for semantic segmentation, available at: https://arxiv.org/pdf/1909.11065.pdf

[11] Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Stachniss C and Gall J 2019 SemanticKITTI: a dataset for semantic scene understanding of lidar sequences. In Proc. of the IEEE Int. Conf. on Computer Vision 9297 DOI: 10.1109/ICCV.2019.00939

[12] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R and Schiele B 2016 The cityscapes dataset for semantic urban scene understanding. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition 3213 DOI: 10.1109/CVPR.2016.350

[13] Milioto A, Vizzo I, Behley J and Stachniss C 2019 Rangenet++: fast and accurate lidar semantic segmentation. In 2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 4213 DOI: 10.1109/IROS40897.2019.8967762

[14] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition 770 DOI: 10.1109/CVPR.2016.90

[15] Zhu X, Zhou H, Wang T, Hong F, Ma Y, Li W, Lin D 2020 Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation, available at: https://arxiv.org/pdf/2011.10033.pdf

[16] Tang H, Liu Z, Zhao S, Lin Y, Lin J, Wang H and Han S 2020 Searching efficient 3d architectures with sparse point-voxel convolution, available at: https://arxiv.org/abs/2007.16100