





# Power Line Corridor LiDAR Point Cloud Segmentation Using Convolutional Neural Network

Jisheng Yang , Zijun Huang, Maochun Huang, Xianxian Zeng, Dong Li ,  
and Yun Zhang

Guangdong University of Technology, Guangdong, China  
dong.li@gdut.edu.cn

**Abstract.** Regular inspection is important for ensuring safe operation of the power lines. Point cloud segmentation is an efficient way to carry out these inspections. Most of the existing methods depend on priori knowledge from a particular power line corridor, which is not applicable for other unknown power line corridors. To address this problem, we propose the first end-to-end deep learning based framework for power line corridor point cloud segmentation. Specifically, we design an effective channel presentation for Light Detection and Ranging (LiDAR) point clouds and adapt a general convolutional neural network as our basic network. To evaluate the effectiveness and efficiency of our method, we collect and label a dataset, which covers a 720,000 square meter area of power line corridors. To verify the generalization ability of our method, we also test it on KITTI dataset. Experiments show that our method not only achieves high accuracy with fast runtime on power line corridor dataset, but also performs well on KITTI dataset.

**Keywords:** Power line inspection · Point cloud segmentation · Convolutional neural network

## 1 Introduction

Power line is considered as one of the most significant infrastructures, which requires regular inspection to ensure the safe operation of a power grid. Thus, power line components such as power lines and pylons need regular checking to diagnose faults, for example, mechanical damage. In addition, power lines' surrounding objects like trees also require regular inspection in case their branches get close or touch the power line, which will cause disaster. Therefore, it is necessary to develop an automatic method for detecting obstacles along power line corridors. Point cloud segmentation is an efficient approach to carry out power line inspection, which has received much attention [2–6, 9].

---

The first author is a student.

Developing efficient and robust LiDAR point cloud segmentation for power line corridor scene remains as a challenging task owing to the variability of power line corridors. More specifically, power line corridors have complicated terrains where steep slopes and flat grounds interlace, leading the challenge of recognizing objects from unseen and diverse geographical environments. In addition, LiDAR point clouds have variant attribute. Their LiDAR intensity distributions could change dramatically even though they are collected from the same areas, due to the difference of airborne laser scanner’s flight height and atmosphere conditions.

Although researchers have explored point cloud segmentation on power line corridor scene, their works [2–4, 6, 9] depends on handcrafted features, which requires much priori knowledge. Therefore, developing a feasible approach that is able to handle the aforementioned challenges for point cloud segmentation on power line corridor scene remains to be unsolved. Recently, deep convolutional neural networks for point cloud segmentation [1, 7, 15] have been brought into being, but there is a lack of deep learning method for power line corridor point cloud segmentation. Due to this fact, we contribute a deep learning approach for segmenting point cloud on power line corridor scene.

In this paper, we propose an end-to-end pipeline for power line corridor point cloud segmentation. To be more specific, we design an effective channel presentation for LiDAR point cloud and utilize a current state-of-the-art network [1] as our basic network, which is further adapted to be suitable for our input and output channels. To evaluate the effectiveness and efficiency of our approach, we collect and label a large scale point cloud dataset on power line corridor scene to do experiments. To verify the generalization ability of our approach, we also implement it on KITTI dataset [16].

The key contributions of this paper are: (1) It is the first deep learning based approach for segmenting power line corridor point clouds; (2) We design an effective channel presentation for LiDAR point cloud, which is not only suitable for power line corridor scene, but also other scene.

## 2 Related Work

### 2.1 Object Segmentation on Power Line Corridor Scene

Methods for recognizing and extracting objects of power line corridor scene can be divided into two categories: (a) point cloud based methods [2–4]; (b) image based methods [5, 6, 9]. Previous methods primarily work on LiDAR point clouds and comprise multiple stages including calculating hand-crafted features for points, designing filters based on the features and extracting objects with the filters. [4] classifies ground and non-ground points by statistically analyzing the skewness and kurtosis of the LIDAR intensity data and detects power lines by employing Hough transfer after that. [2] calculates 21 features of points, and use them to train a decision tree based filter, which is used to segment points. [3] removes ground by applying elevation-difference and slope criteria and then use a combination of height and spacial-density filters to extract power lines. For the methods mentioned above, the selection of handcrafted features highly depends

on priori knowledge from specific areas, which makes them hard to generalize and infeasible in practical applications. Recently, many researchers utilize unmanned aerial vehicle (UAV) images to carry out power line inspection [5, 6, 9]. [6] segments power line components with three steps. It first segments images with fully convolutional neural network, and then implements 3D reconstruction with the images to get a point cloud, and finally matches points from images to the point cloud. In this case, compounded errors are caused by multiple steps. [9] proposed a novel method to sidestep the 3D reconstruction. It first utilize UAV images and ground sample distance (GSD) to generate Epipolar images and then uses left and right Epipolar images to calculate the 3D vectors of power lines. This method extracts power lines in a single step and achieve high accuracy, but its performance is sensitive to the illumination and resolution of the images.

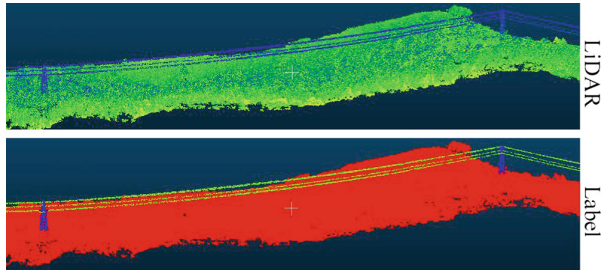
## 2.2 3D Point Cloud Semantic Segmentation

Previous methods depends on intrinsic [10, 11] and extrinsic [12] hand-crafted features to address specific semantic segmentation tasks. Invariant descriptors make them hard to generalize. Later on, deep learning methods occurs and shows great generalization performance by taking the advantage of massive training data. Volumetric CNNs [13, 14] are the pioneers to implement 3D convolutional neural network on voxelized point cloud inputs. But their power is limited by the sparsity of data presentation and huge computation cost. Recently, light convolutional neural networks [1, 7, 15] was developed, which achieve high efficiency by directly consuming point clouds. But these methods are far from mature and only work well on specific experimental dataset.

# 3 Method Description

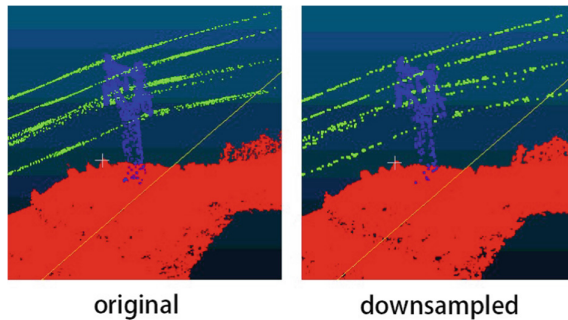
## 3.1 Dataset Collection

**Power Line Corridor Dataset.** To the best of our knowledge, there is no official or public point cloud dataset on power line corridor scene. In order to conduct our study, we collect and label a large scale power line corridor dataset. The original point cloud data is obtained by airborne laser scanners. And we manually label the point clouds by a software named CloudCompare [19]. Points of the dataset are classified into nine categories – high-voltage iron pylon, columnar pylon, power line, lightning protection line, insulator, tree, ground, others and noise. Finally, we get a point cloud dataset with 16 files, which covers a 720,000 m<sup>2</sup> area of power line corridors in total, containing eight kinds of different pylons. In this dataset, there are variable terrains where steep slopes and flat grounds interlace with each other. In addition, the distributions of LiDAR intensity varies from one file to another, because flight height and atmosphere conditions could be different during each journey of the airborne laser scanner. In order to simplify the segmentation task but also take practical applications into consideration, in experiments, we rearrange our dataset into three categories—pylon, power line and others. Others category mainly contains trees and a small part of grounds. One example of the rearranged dataset is shown in Fig. 1.



**Fig. 1.** Examples of LiDAR point cloud and segmentation label. LiDAR point cloud is at the top. The segmentation label is at the bottom. Polygons are denoted in blue, power lines in yellow and trees in red. (Color figure online)

**KITTI Point Cloud Dataset.** Initial data is from KITTI [16] Velodyne point clouds, which is an autonomous driving scene. However KITTI does not provide the point-wise labels. Due to the fact, we label the data by ourselves with the method described by [17]. Specifically, using the 3D bounding box labels from KITTI, all points within a 3D bounding box are considered belonging to an object. Corresponding label is then assigned to each point. With this method, we collected 7481 point clouds with point-wise labels.



**Fig. 2.** Visualization of the original and downsampled point cloud Examples. Original point cloud example is on the left. Downsampled point cloud example is on the right. (Color figure online)

### 3.2 Point Cloud Preprocessing

In order to feed point clouds into the CNN based model, we need to preprocess the point clouds into a series of groups that meets the need of CNN's input. In specific, point clouds are cut into several  $W \times W$  m<sup>2</sup> square blocks with identical area. And then,  $N$  points are randomly sampled from each block. So points in each block are of size  $N \times C$ , where  $C$  represents channels. In this way, points in

blocks are of the same size, so the CNN based model can directly consume them. When preprocessing power line corridor point cloud dataset,  $W = 10$ ,  $N = 4096$  is an appropriate choice.  $W = 10$  is selected because blocks of such size are able to cover most areas of the components of interest like pylons and trees in the power line corridor. By statistics, pylons' widths are in range of 8–11 m and trees' are less than 6 m.  $N = 4096$  is selected because downsampling point clouds to 40.96 points/m<sup>2</sup> not only still let point clouds completely describe the profiles of primary components in the power line corridors, as shown in Fig. 2, but also simplify computation.

### 3.3 An Effective Channel Presentation for LiDAR Points

Unlike [1, 7, 15] which try to explore more complicated and powerful network architectures, this paper aims to find an ideal channel presentation for LiDAR point clouds. The channel presentations p0–p4 that we will discuss are defined as follows:

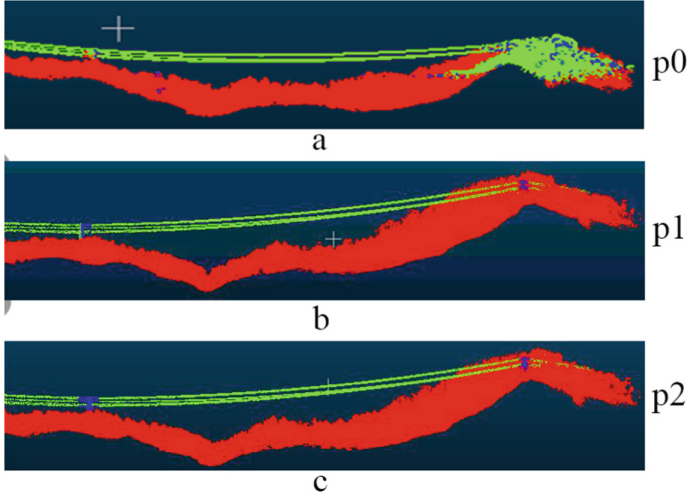
$$\begin{aligned} p0 &: X_c, Y_c, Z, I_N, X_n, Y_n, Z_n \\ p1 &: X_c, Y_c, Z_l, I_N, X_n, Y_n, Z_n \\ p2 &: X_c, Y_c, Z_l, I_{N2}, X_n, Y_n, Z_n \\ p3 &: X_c, Y_c, Z_l, I_{N1}, X_n, Y_n, Z_n, I_{N2} \end{aligned}$$

$$\begin{aligned} F_c &= F - (F_{min\_of\_block} + block\_size/2) \\ F_N &= F/255.0, \quad F_{N1} = F/F_{max\_of\_block}, \quad F_{N2} = F/F_{max\_of\_file} \\ F_n &= F/F_{max\_of\_file}, \quad F_l = F - F_{min\_of\_block} \end{aligned}$$

where  $F_{min\_of\_block}$  is the minimal value of feature  $F$  within a block;  $F_{max\_of\_file}$  is the maximal value of feature  $F$  within the current point cloud file; digit 255.0 is the maximal value of LiDAR intensity;  $block\_size$  is the width of the block.

Under presentation p0 that is directly transformed from PointNet's, pylon category cannot be recognized by the model unless weighted-loss is used during training. Note that only this experiment uses weighted-loss, the rest of experiments in this paper are done without weighted-loss. The result trained with weighted-loss under p0 is shown in Fig. 3a. As seen from the result, points on the flat region can be quite precisely recognized, however, points on the top of the slope are severely misclassified as pylons or power lines. From this phenomenon, it can be inferred that the model tends to classify points relying on global height information—altitude, causing points with the same altitude to gain the same classification labels. Hence, the model needs local height information to make prediction. From intuitional perspective, within a block, pylons should be higher than the ground or trees, and the power lines should locate in the upper part of pylons. In presentation p0, both feature 'Z' and 'Z<sub>n</sub>' represent global height information. So we design p1 to replace the third feature 'Z' with

$'Z_l = Z - Z_{min\_of\_block}'$ , which represents information of local height. Now we get the first three coordinate features for local geometric information, and the last three coordinate features for global. With the new presentation p1, we obtain predicted result as shown in Fig. 3b. From the picture we see that pylon category is recognized and there are no longer a large amount of misclassified points at the top of the slope, which proves our new presentation more effective.



**Fig. 3.** Predicted results under different channel presentations. Figure 3a is the result under presentation p0. Figure 3b is the result under presentation p1. Figure 3c is the result under presentation p2. (Color figure online)

In addition, we notice that distributions of LiDAR intensity in different point cloud files of our power line corridor dataset are quite different, as shown in Fig. 4, and list different point clouds' maximal value of LiDAR intensity in Table 1. As mentioned above, presentation p1 normalize intensity with a fixed value 255.0. That is not optimal, because intensity features are more obvious if normalized intensity values have relatively bigger contrast among points. So we design p2 to normalize the intensity feature with the maximal intensity value of the current point cloud file rather than a fixed value. So the normalized  $'I_N'$  is replaced with  $'I_{N2} = I/I_{max\_of\_file}'$ . Through this modification, we obtain further improvement in the predicted result as shown in Fig. 3c. Finally, following the concept of local and global features, the existing normalized intensity feature  $'I_{N2} = I/I_{max\_of\_file}'$  is regarded as a global feature and we design p3 by adding an extra feature  $'I_{N1} = I/I_{max\_of\_block}'$  as a local intensity feature. With presentation p3, we obtain the best performance among our experiments.

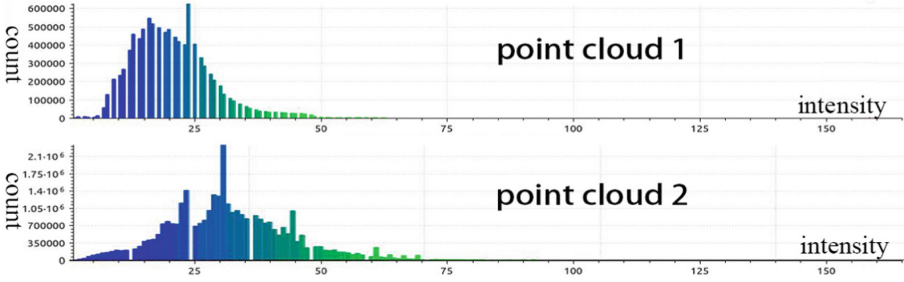


Fig. 4. Distribution of laser intensity for different point clouds

Table 1. Maximal intensity value of different point clouds

Point cloud index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Maximal intensity	147	100	101	206	100	163	236	166	249	120	101	150	100	112	126	101

### 3.4 Network Structure

Our deep learning based framework is shown in Fig. 5, which is adapted from PointNet [1]. It is a simple but effective 3D semantic segmentation network that directly consumes point clouds.

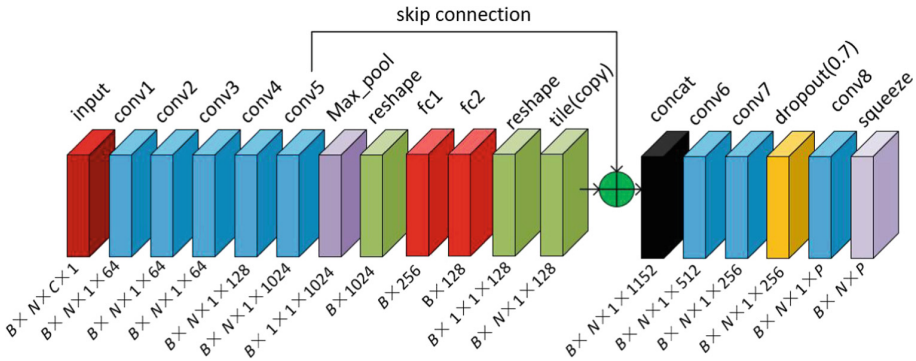


Fig. 5. Network architecture (Color figure online)

The input of the network is a  $B \times N \times C \times 1$  tensor representing points within a block as described in Sect. 3.2, where  $B$  is the batch size;  $N$  is the amount of points sampled from a block;  $C$  represents channels, which is mentioned in Sect. 3.3; and digit 1 is the expended dimension that helps to form the input shape for a CNN. Conv1~conv5 layers implement point-wise convolutions, that is to say each point is convolved independently. These five layers is used to extract features from each point. Next, a symmetry function, max pooling is

applied to make the model invariant to input permutation, dealing with the unordered point clouds. Skip connection is then used to fuse local features from conv5 and global features from fc2. Fusing lower level and higher level features can effectively improve smoothness and detail of the segmentation output [8]. Finally, the model gets predicted classification for each point. In the last layer,  $P$  is the number of categories.

## 4 Experiments

The effectiveness and efficiency of our approach are evaluated on power line corridor dataset. In addition, the generalization ability of our approach is verified on KITTI dataset.

### 4.1 Evaluation Metrics

Our method’s performance is evaluated on class-level. We compare predicted and ground-truth label values and calculate precision, recall and IoU (intersection over union) between them respectively. Among the three, IoU is the most important metric to estimate the performance of segmentation. So we mainly discuss IoU in this section. The evaluation metrics are defined as follows:

$$Precision_c = \frac{|P_c \cap G_c|}{|P_c|}, recall_c = \frac{|P_c \cap G_c|}{|G_c|}, IoU_c = \frac{|P_c \cap G_c|}{|P_c \cup G_c|} \quad (1)$$

where  $P_c$  and  $G_c$  represent the predicted and ground truth point sets belonging to class  $c$  respectively,  $|*|$  calculates the amount of points in the point set  $*$ .

### 4.2 Segmentation on Power Line Corridor Dataset

**Settings.** There are 16 large scale point cloud files in our dataset. Each of them covers an approximate area. One file is randomly chosen as validation set and the remaining 15 files are set as training set. In our split, validation set is ensured to be unseen during training. Note that our dataset is reorganized into three categories: pylon, power line and others. Others category mainly contains trees and grounds. In point cloud preprocessing procedure, point clouds are separated into  $10 \times 10 \text{ m}^2$  blocks and 4096 points are randomly sampled from each block as the inputs, as discussed in Sect.3.2. And then, the network mentioned in Sect. 3.4 is trained without weighted-loss on a GTX 1070 GPU.

**Results and Analysis.** Experiments are implemented under different channel presentations that are discussed in Sect.3.3. We compare our method with PointNet [1]. The results are summarized in Table 2. Note that results under p0 represent the results of PointNet.

Local height channel is critical. Point presentation of PointNet does not contain local height channel. As we can see, PointNet fails to recognize pylon category. In contrast, presentation p1 contains local height channel. Under p1, model



**Table 2.** Experiment Results under Different Point Channel Presentation. All values in the table are in percentages

	Pylon				Power line				Others			
	p0	p1	p2	p3	p0	p1	p2	p3	p0	p1	p2	p3
Precision	Nan	87.41	88.65	85.53	95.55	90.88	98.72	99.37	99.87	99.99	99.99	99.99
Recall	0	51.11	81.21	89.29	97.75	99.39	99.11	98.69	99.99	99.93	99.99	99.99
IoU	0	47.61	73.56	77.57	94.30	90.37	97.85	98.07	99.94	99.94	99.99	99.99

can segment pylon category with a relatively high IoU 47.61, which shows that local height channel is important and proves our channel presentation for LiDAR point cloud effective.

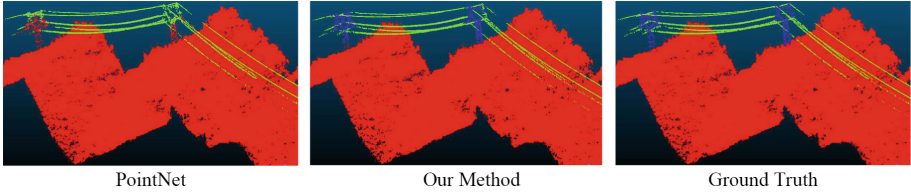
Appropriate normalization for LiDAR intensity is important. Presentation p1 normalizes intensity with a fixed value, while p2 normalizes intensity with a flexible value. This modification lets normalized intensity values have relatively bigger contrast among points, making points have more obvious intensity features. From the results, a relative 54.5% IoU improvement from presentation p1 to p2 in pylon category can be seen and IoUs of other categories also improve, which shows that normalizing features properly is vital. In addition, following the concept of local and global features, presentation p3 adds another normalized intensity feature as a local feature while regarding the original one as a global feature. Presentation p3 further improves the IoU in every category and it performs the best in our experiments.

**Table 3.** Statistics on our power line corridor dataset. All the values in this table are in percentages.

	Point percentage	Block percentage
Pylon	0.14	0.80
Power transmission line	4.15	25.89
Others	95.71	98.71

Note that recalls for both power line and others categories are near perfect, higher than 98%, which is desirable for power line inspection, as dangerous spots lie where trees are too close to power lines. However, even when using the most effective point presentation p3, recall for pylon is 89%, though high but relatively lower than other two categories. Lower performance on pylon category is attributed to two reasons: (a) Samples of different categories are unbalanced. Quantity characteristics of the power line corridor dataset is summarized in Table 3. Both point and block percentage on pylon category are much smaller than those on other two categories. Point percentage is defined as (number of points belonging to category  $c$ /number of all points); Block percentage is defined as (number of blocks containing category  $c$  points/number of all blocks). As a result, model does not ‘see’ enough samples of pylons so has lower performance

of segmenting them; (b) As shown in Fig. 2, geometric structure of pylon is more complicated than those of other two categories and pylons have joint places with power lines and grounds, making it more difficult to segment.



**Fig. 6.** Visualization of PointNet’s and our method’s predicted results. (Color figure online)

A visualization of the segmentation results by our method, PointNet and ground truth label is shown in Fig. 6. Our method performs well on all categories while PointNet cannot recognize pylon category. On the whole, our predicted result is very close to the ground truth label. In addition, our model is efficient. On a GTX 1070 GPU, it only takes 0.93 ms to predict 1 m<sup>2</sup> data of our power line corridor dataset.

### 4.3 Segmentation on KITTI Point Cloud Dataset

**Settings.** KITTI point cloud dataset is randomly separated into a training set with 6481 frames and a validation set with 1000 frames. In addition, each point cloud is separated into  $4 \times 4$  m<sup>2</sup> blocks. And 2048 points are randomly sampled from each block as the inputs. Reasons for the setting here is similar to those of the power line corridor dataset. Our method is implemented under our presentation p3 mentioned above, and PointNet [1] is under its original presentation p0. The model is trained on the same machine mentioned above.

**Results and Analysis.** To evaluate the generalization ability of our method, experiments on KITTI point cloud dataset are included. We compare our method with PointNet and SqueezeSeg [17]. SqueezeSeg is a leading segmentation algorithm on KITTI point cloud dataset. Experiment results on IoU are summarized in Table 4. Our method surpasses PointNet on all categories and is comparable with SqueezeSeg on car and pedestrian categories. IoU on car category from our method is even higher than that from SqueezeSeg without CRF. Consequently, our method not only has prominent performance on power line corridor dataset, but also generalizes well on KITTI dataset.

Note that PointNet’s performance on KITTI is much better than that on Power Line Corridor Dataset. The difference is attributed to the following reason. PointNet was initially designed on S3DIS dataset [18], an indoor scene with

**Table 4.** Comparison with other methods on KITTI Dataset. Metric is IoU. All values in the table are in percentages

	Car	Pedestrian	Cyclist
SqueezeSeg	60.9	22.8	26.4
SqueezeSeg with CRF	64.6	21.8	25.1
PointNet	60.7	13.9	8.4
Our method	64.1	21.8	8.7

rooms. And KITTI is collected from roads. Both rooms and roads are flat landforms, which are different from power line corridor where steep slopes exist. In such conditions, the local height information referring to a block could be represented by global height referring to a file, so PointNet’s channel presentation is effective on KITTI. This experiment also shows that our channel presentation is suitable for various terrains.

## 5 Conclusion

We propose the first deep learning based framework for power line corridor point cloud segmentation. In specific, we design an effective channel presentation for LiDAR point clouds and adapt a general convolutional neural network as our basic network. Compared to the existing works, our approach has three superiorities: (a) It does not rely on hand-crafted features which requires priori knowledge, but utilize a trainable deep learning model, which has strong generalization ability; (b) It does not use images but LiDAR data, so it is not sensitive to outer factors like illumination and thus more stable; (c) It finishes segmentation tasks within a single stage rather than multiple stages, so is able to leverage object context and get rid of propagated errors. In order to evaluate the effectiveness of our approach, we collect and label a large scale point cloud dataset of power line corridor scene to do experiments. To verify the generalization ability of our approach, we also test it on KITTI dataset. Experiments shows that our model can achieve very high segmentation accuracy with fast and stable runtime ( $0.93 \pm 0.5$  ms/m<sup>2</sup>) on power line corridor dataset and generalize well on KITTI, which means that our approach is not only potential to meet the urgent need of automatically inspecting power lines but also is useful for other applications like autonomous driving.

## References

1. Qi, C.R., et al.: PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

2. Kim, H.B., Sohn, G.: 3D classification of power-line scene from airborne laser scanning data using random forests. *Int. Arch. Photogramm. Remote Sens.* **38**, 126–132 (2010)
3. Guan, H., et al.: Extraction of power-transmission lines from vehicle-borne LiDAR data. *Int. J. Remote Sens.* **37**(1), 229–247 (2016)
4. Liu, Y., et al.: Classification of airborne LiDAR intensity data using statistical analysis and hough transform with application to power line corridors. In: 2009 Digital Image Computing: Techniques and Applications. IEEE (2009)
5. Zhang, Y., et al.: UAV low altitude photogrammetry for power line inspection. *ISPRS Int. J. Geo-Inf.* **6**(1), 14 (2017)
6. Maurer, M., et al.: Automated inspection of power line corridors to measure vegetation undercut using UAV-based images. *ISPRS Ann. Photogrammetry Remote Sens. Spat. Inf. Sci.* **4** (2017)
7. Qi, C.R., et al.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems* (2017)
8. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015)
9. Zhang, Y., et al.: Automatic power line inspection using UAV images. *Remote Sens.* **9**(8), 824 (2017)
10. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: a quantum mechanical approach to shape analysis. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE (2011)
11. Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE (2010)
12. Rusu, R.B., et al.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE (2008)
13. Maturana, D., Scherer, S.: VoxNet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2015)
14. Qi, C.R., et al.: Volumetric and multi-view CNNs for object classification on 3D data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016)
15. He, T., et al.: GeoNet: deep geodesic networks for point cloud analysis. arXiv preprint [arXiv:1901.00680](https://arxiv.org/abs/1901.00680) (2019)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In: *Proceedings CVPR* (2012)
17. Wu, B., et al.: SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2018)
18. Armeni, I., et al.: 3D semantic parsing of large-scale indoor spaces. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016)
19. <https://www.danielgm.net/cc/>