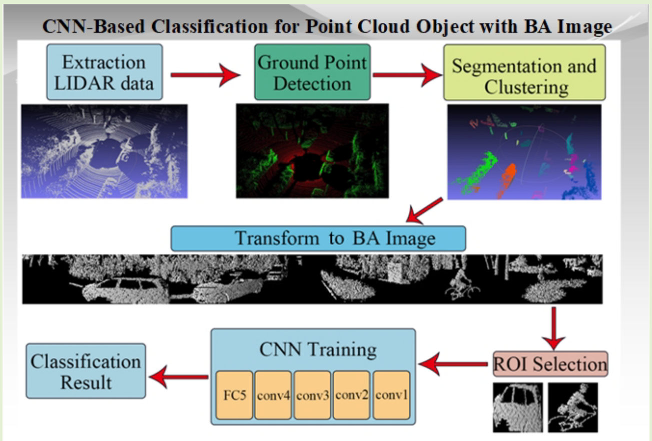
CNN-Based Classification for Point Cloud Object With Bearing Angle Image

Chien-Chou Lin, *Member, IEEE*, Chih-Hung Kuo, *Member, IEEE*, and Hsin-Te Chiang

***Abstract*—Convolutional neural network (CNN), one of the branches of deep neural networks, has been widely used in image recognition, natural language processing, and other related fields with great success recently. This paper proposes a novel framework with CNN to classify objects in a point cloud captured by LiDAR on urban streets. The proposed BA-CNN algorithm is composed of five steps: (i) removing ground points, (ii) clustering objects, (iii) transforming to bearing angle images, (iv) ROI selection, and (V) identifying objects by CNN. In the first step, ground points are removed by the multi-threshold-based ground detection to reduce the processing time. Then, a flood-fill-based clustering method is used for object segmentation. Those individual point cloud objects are converted to bearing angle (BA) images. Then, a well-trained CNN is used to classify objects with BA images. The main contribution of this paper is proposing an efficient recognition method that uses the information from point clouds only. In contrast, because most 3D object classifiers use the fusion of point clouds and color images, their models are very complicated and take a colossal amount of memory to store the parameters. Since the ground point detection and object clustering process all points along with the scanline-major order and layer-major order, the proposed algorithm performs better in terms of time consumption and memory consumption. In the experiment, three scenes from KITTI dataset are used for training and testing the proposed BA-CNN classifier, and the proposed BA-CNN achieves high classification accuracy.**



***Index Terms*—Bearing angle image, instance segmentation, object classification, convolutional neural network (CNN), light detection and ranging (LiDAR), point cloud segmentation.**

I. Introduction

With 3D surface measurement devices have been widely used, much research about 3D modeling and 3D object recognition/classification was proposed. 3D modeling is to align partial overlapping several point clouds with variant viewpoints into a standard coordinate system. With optimal translation and rotation, these point clouds can reconstruct the object's 3D shape. As for the 3D object recognition/ classification, individual objects would be separated and recognized with their partial surface from a point cloud.

Since road scenes can be scanned easily by LiDAR sensors equipped on a vehicle and represented as point clouds, object recognition/classification is essential for the Advanced Driver Assistance System (ADAS). Basically, ADAS does not replace the role of a driver in car control but rather assists him in obtaining information on vehicle operation and its surrounding environment. In ADAS, object recognition usually relies on computer vision methods involving variant sensors, including cameras, sonars, radars, LiDAR (Light Detection and Ranging), and so on.

Computer vision techniques in ADAS fall roughly into two categories: 2D image-based approaches and 3D data-based approaches. 2D image-based approach is prevalent in the traditional computer vision of ADAS because it is more cost-effective than other sensors. However, the quality of the images may be affected by other factors, such as uneven

Manuscript received October 5, 2021; accepted November 19, 2021. Date of publication November 23, 2021; date of current version December 29, 2021. This work was supported in part by the Intelligent Recognition Industry Service Center (IRIS) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan; and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2221-E-224-054 and Grant MOST 107-2221-E-224-050. The associate editor coordinating the review of this article and approving it for publication was Dr. Brajesh Kumar Kaushik. *(Corresponding author: Chien-Chou Lin.)*

Chien-Chou Lin is with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Douliu, Yunlin County 64002, Taiwan (e-mail: linchien@yuntech.edu.tw).

Chih-Hung Kuo is with the Department of Electrical Engineering, National Cheng Kung University, Tainan City 70101, Taiwan (e-mail: chkuo@ee.ncku.edu.tw).

Hsin-Te Chiang was with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Douliu, Yunlin County 64002, Taiwan. He is now with Mirle Automation Corporation, Hsinchu City 30076, Taiwan (e-mail: a3859484g@gmail.com).

Digital Object Identifier 10.1109/JSEN.2021.3130268

illumination, leading to massive false alarms. On the other hand, the 3D image is usually captured through LiDAR, which emits laser and measures the time of its reflection, and hence it is more accurate and reliable. Furthermore, unlike 2D flat data, which are projections of the 3D environment, the 3D images can keep the spatial information of surface features. LiDAR sensors generate point clouds that contain a much larger scale of data than 2D color images. Dimension reduction is required to improve the performance.

Some 3D object identifiers have recently attempted to fuse point clouds and color images, which need to calibrate the coordinate systems of multiple sensors and merge into a unified coordinate system. Since point clouds and color images are used, the RGB-D-based approaches have better accuracy than others. However, the main drawback of these approaches is time-consuming for the fusion of various resolutions in sensors.

This paper aims to efficiently and correctly classify objects solely based on point clouds into three classes: cars, pedestrians, and street clutter. It is a great challenge to recognize point cloud objects with CNN since the 3D features of the point cloud are different from 2D images, and the disordered point clouds cannot be used in CNN directly. The primary contributions of this paper could be summarized as follows:

- We propose a novel 3D object detection algorithm (BA-CNN) by using conventional CNN models.
- The proposed method converts the point cloud to bearing angle images which preserves the object's shape and surface features exactly.
- We performed ablation studies to examine the key factors that can increase the accuracy of BA-CNN.
- Comparing with the state-of-the-art approaches, the accuracy of the BA-CNN is better than others that are solely based on LiDAR point clouds.

The rest of this paper is organized as follows. The related works of object registration are reviewed in Section 2. In Section 3, we propose a CNN-based object classifier. Section 4 shows and discusses the experimental results. Section 5 concludes this paper.

II. Related Work

In recent years, object recognition by using laser scanning data has been widely discussed in many works of literature. Object recognition aims to identify objects in a scene correctly. However, due to noise and occlusions, object recognition is a challenging task. In general, object recognition consists of several essential steps: segmentation, feature extraction, and classification. In a crowded scene, the point cloud is firstly segmented into background and foreground points. To separate foreground 3D points into several individual groups, some features, e.g., edge/border of points; normal of points, etc., are used for segmentation. The points within the same boundaries are grouped into the same object. The similarities of regions, e.g., normal of KNN, slope, and distance, are also used for grouping the points.

Indeed, the classifiers depend on the input types, which are the features extracted from the individual objects. According to the input features, the classification can be roughly categorized into local feature-based approaches and global feature-based approaches. The local feature is extracted from an interesting point and its neighbors, such as normal points, FPHF, PCA, etc. In [1], Guo, *et al.* proposed the TriSI feature, which represents the three orthogonal coordinate axes of the Local Reference Frame (LRF) by using the implicit geometrical information of neighboring triangular faces to recognize 3D objects. Bariya *et al.* encoded the scale variability of the surface geometry by an interpretation tree for each object [2]. The tree nodes are the object features and contain a hypothesis formed by the feature correspondences at that node and all its parent nodes.

While the local feature-based methods use the 3D features derived by a point and its neighbors, the global feature-based approaches use the features extracted from the whole objects for recognition, e.g., 2D contours or 3D voxels. Since more features can be obtained from the surface of an object, the global feature-based method is usually more robust than point-feature-based methods. Another advantage is that the global features can significantly reduce the data volume to improve the performance of the classifiers [3]–[5].

In [3], the foreground objects were detected by subtracting two consecutive frames. The differences were clustered by connected component labeling (CCL). Then, the moving objects, the foreground, were classified by their aspect ratios. This method's main drawback is that it can only be applied for a sequence of scanning frames. In [5], an appearance-based identifier projected point cloud objects onto a 2-dimensional plane and extracted shape features of the 2D images. Since the proposed classifier used an SVM classifier trained by 2D features, some important 3D features might be discarded, leading to unstable recognition. In [7], three types of features, local descriptor histograms (LDHs), spin images, and general shape and point distribution features, were used to classify roadside objects. LDHs and spin images were applied for SVM based classifier. In [8], Yang also used an SVM-based classifier.

In self-driving applications [9]–[12], the scene understanding which goal is classifying on-road objects, e.g., pedestrians, cars, bicycles, and buildings, is essential. In [13], objects were recognized by shapes extracted from partial 3D point clouds for the localization function of automated guided vehicles (AGVs). In [14], [15], supervoxel-based methods extract 3D road objects, including road boundary and light-pole. Supervoxels are generated by removing background points and clustering volumetric over-segmentations into supervoxels. In [16], the tracking and motion estimation method of obstacles using just a 3-D point cloud was proposed. In [9], a classifier using multi-scale stereo pixels as the features was proposed. A multi-layer LiDAR captured point clouds, and the density of point clouds decreased with the distance from the sensor. Therefore, it is challenging to detect objects far away because few features are extracted from sparse points.

Recently, convolutional neural networks (CNN) have been used in image recognition widely, and some works of literature have adopted CNN as a classifier [17]–[19]. The CNN-based

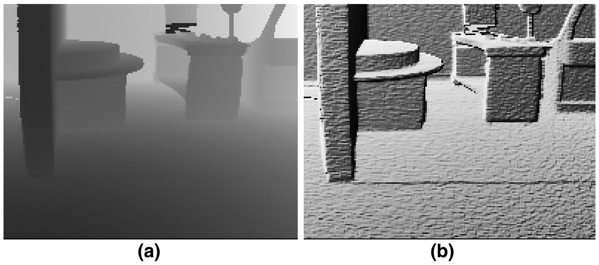


Fig. 1. (a) Depth image and (b) bearing angle image, colors represent different depths.

architectures are potential in segmenting and classifying point cloud data. For example, in [20], Börcs *et al.* proposed a CNN-based object classifier that transforms 3D objects to depth images. In [21] and [22], a Fully Convolutional Network (FCN) was adopted for segmentation and classification for an image containing multiple objects. In [23], PointNet, a modified CNN, used density occupancy grids as internal representation to classify objects. However, because no data structure is suitable for presenting the geometric relations of points, the point cloud is hard to process for CNN directly. Therefore, transforming point clouds into 2D images or voxelizing them to 3D grids is necessary for CNN-based classifiers.

There are some approaches proposed for transforming point clouds into 2D images. One of the most common methods is depth image, representing a point depth as the grayscale. Such a method preserves only the geometric relations of objects but not the geometric relations of points. The surface details of objects were lost. Another technique uses bearing angle images [10] proposed by Davide *et al.* to represent the point cloud objects. A bearing angle image contains the depth information of the whole three-dimensional object and the depth relationship between each point and its surrounding points. As the example of depth image shown in Fig. 1(a), points of the same object are almost entirely in the same color, which means they lie at the same distance. Fig. 1(b) shows the bearing angle image. It is easy to see that the points of the same plane are almost in the same grayscale, and the surface details are preserved after transformation. Therefore, the bearing angle image is adopted in this paper to secure surface details for recognition. In [24], the BA images were used for recognizing complex indoor scenes. The SURF features of BA images were matched. The authors also used the BA images for robot localization [25].

III. BA-CNN: Bearing Angel Image for Object Recognition by CNN

The proposed algorithm aims to classify objects into three categories: cars, pedestrians, and others (street clutters, facades) using only point clouds. The point cloud obtained by LiDAR mounted on a mobile platform is processed by removing ground points to reduce the points. Then, rest points are clustered and separated into individual objects. Those objects are transformed into bearing angle images which are used to train the convolutional neural network later. The proposed BA-CNN model is adopted as the object classifier. In Fig. 2, the workflow of the proposed algorithm is composed of five steps: (1) removal of ground points, (2) object segmentation, (3) transformation from point cloud data to bearing angle images, (4) ROI selection, and (5) object classification by CNN. The details of those five steps are elaborated in the following sections.

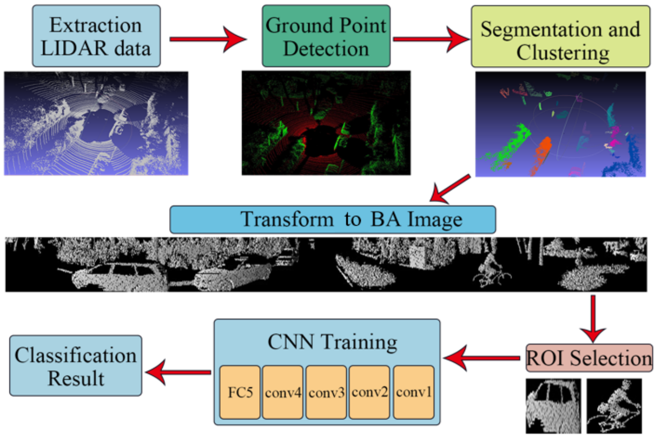


Fig. 2. Flow chart of point cloud classification using CNN.

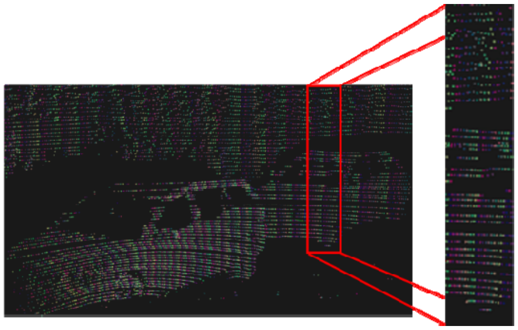


Fig. 3. A scene captured by LiDAR. Points with the same color belong to the same scan line.

A. Data Structure Based on LiDAR Scan Order

Most applications of point clouds need to compute the closest neighbor for a given point with techniques like normal vector estimation, surface simplification, and finite element modeling. The nearest neighbor researches proposed in recent decades used specific data structures. The data structure of the point cloud is an essential factor of performance for finding the nearest points, e.g., KD-tree. Fig. 3 shows a frame captured by Velodyne HDL-64E, a multi-beam LiDAR, 64 layers with 2084 points per layer. Points of the same color belong to the same scan line. A simple data structure to keep the LiDAR scanning order is a two-dimensional array that stores x-y-z coordinates of points. Although it is not convenient for finding the closest neighbors of a point by this data structure, the proposed ground point detection and object segmentation methods can take advantage of simple data structure and process the data by scanline-major order. The proposed algorithm performs well with the simple data

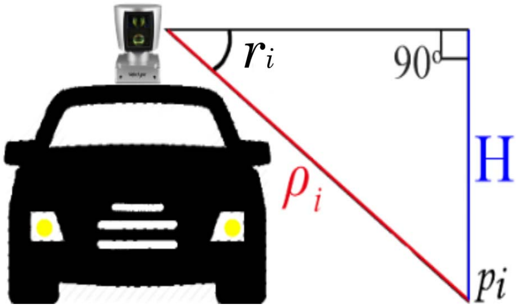


Fig. 4. Corresponding relationship in LIDAR and ground point.

structure by consuming less computation and memory usage than other approaches.

B. Ground Point Detection With Multiple Threshold

Usually, a point cloud has a large amount of 3D points in which ground points account for a large proportion, approximately 25%-50% of the raw data. Most applications of the point cloud use ground point removing to reduce the complexity, such as object classification, dynamic object detection, and so on [5], [6], [20]. In general, approaches of ground segmentations are based on the heights of points [26], [27], the slope of two sequence points [28], [29], or features of areas [30].

The height is one of the essential features for a ground point. Fig. 4 illustrates that the height of a point can be obtained by $(\rho_i \sin\gamma_i + H_0)$, where the extrinsic parameters of LiDAR mounted on a car are known where ρ_i is the measured distance of the point p_i and γ_i $(-24.8^\circ \leq \gamma_i \leq 2^\circ)$ is the vertical angle of the laser beam to the horizon. A positive value γ_i means the laser is pointing up, and a negative value means the laser pointing down. Therefore, a ground point can be detected as

$$|\rho_i \sin\gamma_i + H_0| < H_{th} \tag{1}$$

where the threshold of height is set as 15 cm in this work. Using only the heights of points to detect ground points is not accurate enough since the height of the ground is not constant. For example, on an uphill road, the height of the ground points varies over time. In this paper, the height criterion in (1) is only considered as the initial ground point. We further detect the points from near to far along with two sequent points from -24.8° to 2° for each scan line. It is assumed that the neighboring place around the LiDAR is flat. If the first ground point a is found, whether the adjacent point b is also the ground is determined according to the slope formally defined as

$$S = \frac{|y_a - y_b|}{\sqrt{(x_a - x_b)^2 + (z_a - z_b)^2}}. \tag{2}$$

In order to reduce the complexity, it is simplified as

$$S^2_{(b,a)} = \frac{(y_a - y_b)^2}{(x_a - x_b)^2 + (z_a - z_b)^2} \tag{3}$$

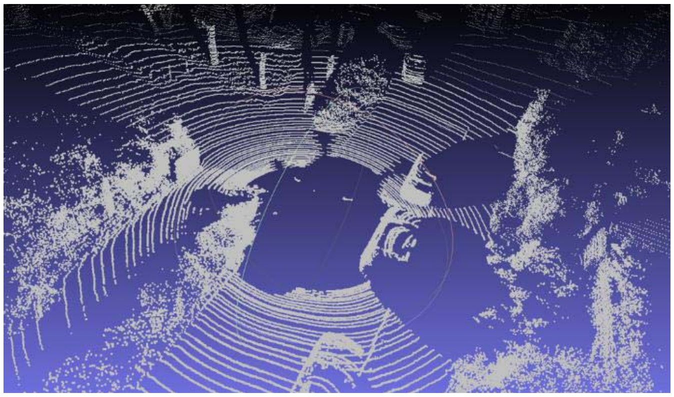


Fig. 5. The raw point cloud. The points near the LiDAR are denser than the area far away from the LiDAR.

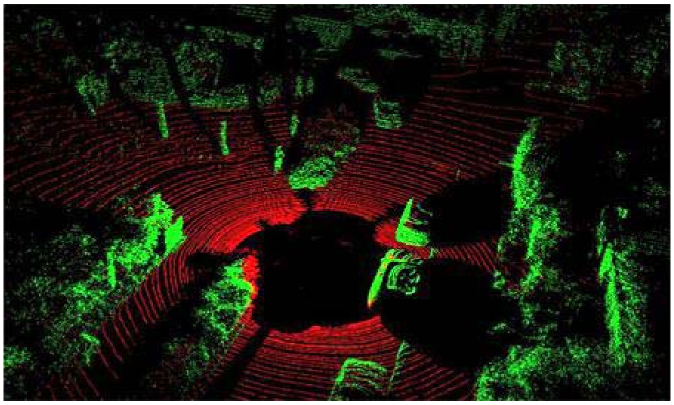


Fig. 6. The result of ground segmentation. Red points are ground points, and green points are object points.

If S is less than the threshold, point b is considered as a ground point. However, S is sensitive to the distance between the two points. In Fig. 5, it is easy to see that the points near the LiDAR are denser than the area far away from the LiDAR. In other words, the distance of two consecutive points in the same scan line is directly proportional to their distance to the LiDAR. Thus, a slight noise of altitude difference in the area near the LiDAR might change the slope significantly. We propose three thresholds for S concerning different distances to make the slope more robust for ground detection. With the distance $d_{(b,a)}$ of any two consecutive points a and b, the slope thresholds are defined as follows:

$$\begin{cases} T = T_0 + \alpha \cdot (d_{near}/d_{(b,a)})^2, & if\ d_{near} \geq d_{(b,a)} \\ T = T_0, & if\ d_{far} \geq d_{(b,a)} \geq d_{near} \\ T = T_0 - \beta(d_{(b,a)}/d_{far})^2, & if\ d_{(b,a)} \geq d_{far} \end{cases} \tag{4}$$

where d_{near} and d_{far} are the predetermined distances of the nearby area and outlying area, respectively. T_0 is the threshold for the points within a middle distance. α and β are constants. In Fig. 6, ground points are segmented in red, and the points of objects are colored in green. After removing ground points, the amount of points is usually reduced to 40% to 60%. Less reserved points make the clustering easier and faster.

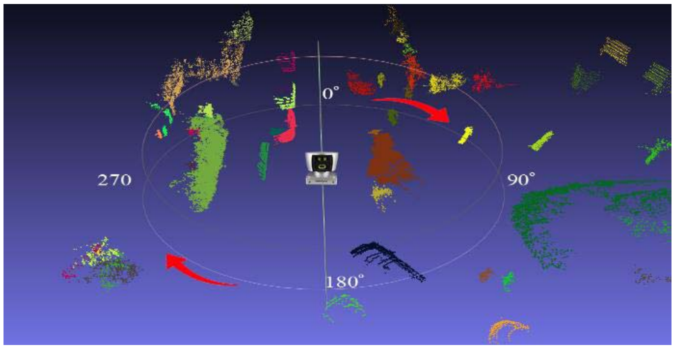


Fig. 7. Clustering the foreground points into several objects.

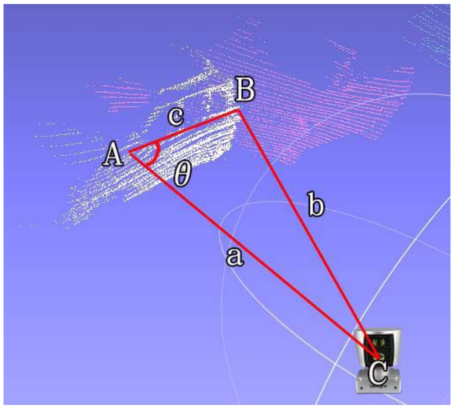


Fig. 8. Corresponding relationship in a triangle. Two adjacent points A and B, and the laser source C.

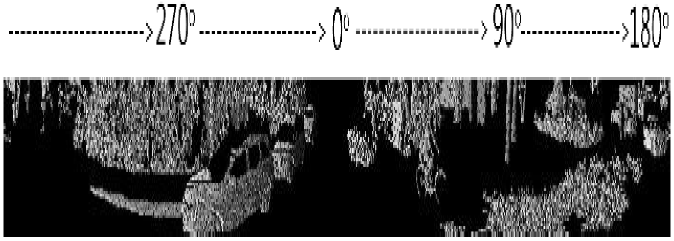


Fig. 9. The BA image of Fig. 7.

Fig. 10. The color image of the same scene of Fig. 9.

C. Object Segmentation Based on Flood-Fill Algorithm

While ground points are removed as background in the previous step, the remaining points are considered as the foreground and have to be grouped into individual objects. Most existing clustering algorithms can be used in grouping point clouds, e.g., k-means, random sample consensus (RANSAC), etc. To cluster the points more efficiently, we adopt the flood-fill algorithm proposed in [11]. Instead of finding the closest points, the flood-fill segmentation uses the nearest neighbors along with scan-line-major order and then layer-major order.

In the first step of the flood-fill algorithm, a new line segment starts from the first and nearest point of a scan line if it is not a missing point or a ground point. If the consecutive point of the same scan line lies within a distance, it is merged into the line segment. Otherwise, the growth of the line segment stops, and those points are marked. The second step is to group those line segments along with layer-major order within a threshold of distance. In Fig. 7, points of different objects are marked in different colors.

D. Transforming Point Cloud to Bearing Angle Image

Since extracting 3D features is more complex than extracting 2D features, point clouds of objects are transformed into 2D images after segmentation. A common method is transforming a 3D point cloud into a range image. However, the gray level of the range image represents only the Z-coordinate of points and neglects the relation between points. To preserve surface details of objects, the bearing angle image of a point cloud of objects is used instead of range images. A pixel of the bearing angle image represents the angle between the laser beam and the vector from the point to a consecutive point. In Fig. 8, there are two adjacent points (A, B) and the laser source C. The bearing angle of A can be obtained by

$$\theta = \cos^{-1}\left(\frac{a^2+b^2-c^2}{2ab}\right), \quad 0 \le \theta \le \pi \tag{5}$$

where a is the measured distance of A and b is that of B. The distance c can be derived by

$$c = \|A(x, y, z) - B(x, y, z)\| \tag{6}$$

To forming a BA image, we convert θ to a gray level by

$$I = \frac{\theta}{\pi} \times 255. \tag{7}$$

Figs 9 and 10 show the obtained BA image from Fig. 7 and the same scene's color image, respectively. Individual object in the point clouds segmented in the previous step is transformed into a BA image used as a training image or a test image for the proposed convolutional neural network (CNN).

E. Recognition by Convolutional Neural Network With BA Image

Since the shapes of BA images of individual objects are different, we set the minimum bounding boxes of individual objects as the regions of interest (ROIs), input to the proposed CNN. The primary goal of this work is to identify cars, pedestrians, and street clutters from candidate ROIs.

The proposed architecture is an AlexNet-like convolutional neural network [31] implemented by the Tensorflow Framework [32]. The model consists of two convolutional layers, two pooling layers, one fully connected layer, and one output layer.

Input grayscale BA images are normalized into a size of 64 × 64. Since the input is similar to the 2D images, 16 filters are used in convolutional layers. Thus, 16 × 64 × 64 feature maps are obtained and passed to the rectified linear unit (ReLU). A ReLU replaces all negative values by zero in the feature maps. The second hidden layer is the max-pooling layer (Pooling). A Pooling layer is adopted to reduce the dimension of each feature map. The result of the previous procedures can be expressed by

$$f(x_i) = LRN(MaxPool(\sigma(W_c x_i))) \tag{8}$$

where x_i is a 64 × 64 BA image and W_c is the convolutional operator. σ is the ReLU activation function defined by

$$\sigma = \max(0, W_c x_i). \tag{9}$$

Then, a pooling layer reduces the dimensions of Max pooling features that use the maximum value of each local cluster of neurons in the feature map. To limit the ReLU activation from increasing the output layer values, a local response normalization (LRN) is used for lateral inhibition. $LRN(\cdot)$ is defined as

$$b^i_{x,y} = a^i_{x,y}/(1 + 10^{-4} \sum_{j=\max(0,j-4)}^{\min(N-1,i+4)} (a^j_{x,y})^2)^{0.75}. \tag{10}$$

The proposed CNN model with two convolutional layers and two pooling layers is expressed by

$$M = (f(f(x_i))). \tag{11}$$

16 × 32 × 32 pooled feature maps are obtained and passed to the fully connected layer (F). The vector s is calculated from the model output M using two fully connected layers:

$$\begin{aligned} s &= F(M) \\ &= F(f_i(f_i(x_i))) \\ &= F(LRN(MaxPool(LRN(MaxPool(\sigma(W_c x_i)))))) \end{aligned} \tag{12}$$

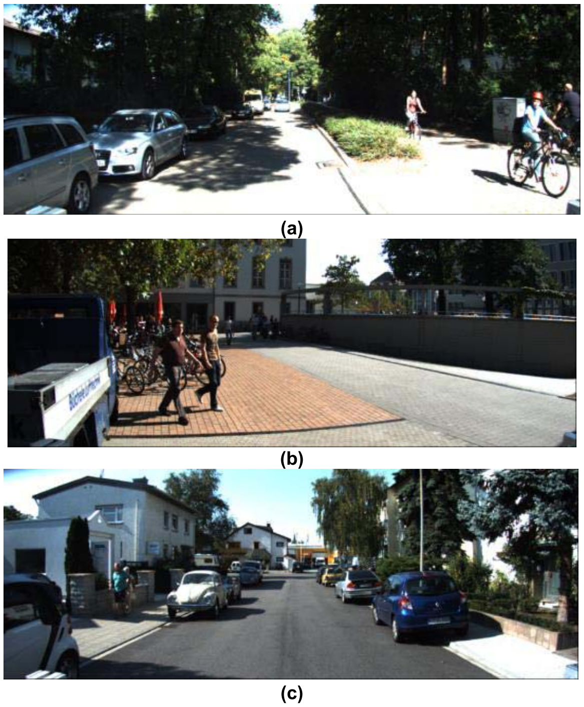
While the purpose of the convolutional and pooling layers is feature extraction, the purpose of the fully connected layer is to learn the classification from the pooled features into three classes. We use Softmax as the activation function in the output layer to ensure the sum of output probabilities is 1.

$$S_i = \frac{e^{S_p}}{\sum_j e^{S_j}} \tag{13}$$

As for the lost function, categorical cross-entropy (CCE) loss is adopted.

$$CCE = \frac{1}{M} \sum_p^M (-\log S_p) \tag{14}$$

Several modified architectures have been experimented with within this paper to obtain the best result of CNN for object classification. The details of their performances are given in the next section.


(a)

(b)

(c)

Fig. 11. (a) Scene 1: Residential area raw data (2011_09_26_drive_0020) from KITTI. (b) Scene 2: Campus (2011_09_28_drive_0043) from KITTI. (c) Scene 3: Residential area raw data (2011_09_26_drive_0035) from KITTI.

IV. Experiments and Analysis

Our experiments take three scenes from the KITTI dataset [17], including raw data of two residential areas (Scene1: 2011_09_26_drive_0020 and Scene3: 2011_09_26_drive_0035) and raw data of one campus (Scene2: 2011_09_28_drive_0043) as shown in Fig. 11. The training set contains 1255 objects, including 500 pedestrians, 221 cars, and 534 street clutters from one residential area and the campus. In our simulation, people riding bikes are regarded as pedestrians. We use the accuracies of classification as performance metrics. The accuracies of different classes and the average accuracy are expressed as follows, respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{15}$$

$$Average = \frac{\sum(TP_i + TN_i)}{\sum(TP_i + FP_i + TN_i + FN_i)} \tag{16}$$

We use C++ to implement the proposed algorithm in Microsoft Visual Studio 2015 with OpenCV 3.2 and Point Cloud Library 1.8.1. The KITTI dataset was obtained by Velodyne HDL-64E LiDAR scanner, which can scan 360 degrees in the horizontal direction and +2 to −24.9 degrees in the vertical direction. Sixty-four scanning lines are distributed across 26.9 degrees in the vertical order, and the maximum measurable distance is 120 meters.

A. Ablation Experiment

To obtain the optimal performance of the proposed algorithm, we focus on input format and network architecture in the ablation study. In the first part of the simulation, we use

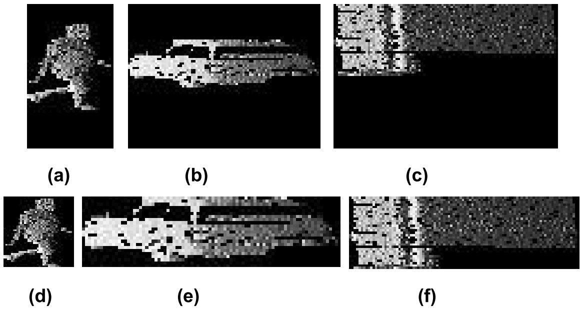


Fig. 12. ROIs with different sizes. (a) Pedestrian within minimal width; (b) vehicle within minimal width; (c) street clutter within minimal width; (d) pedestrian within minimal bounding box; (e) vehicle within minimal bounding box; and (f) street clutter within a minimal bounding box.

two ROI types. One is cropping the target area of a BA image with a fixed height and the object's width, and the other cropped ROI is with a minimal bounding box of the object. As shown in Fig. 12(a)∼(c), the heights of the three former ROIs are the same, but their widths depend on objects. As shown in Fig. 12(d)∼(f), the latter ROIs are the minimal bounding boxes of three types of objects. Then, the image sizes of both types of ROIs are normalized as 64 × 64. In Table I, the first two rows show the accuracy of two different ROIs. The performance of the cropped ROIs with minimal width is better than another one since more contour features are preserved.

In general, the size of the input of CNN affects the computation time significantly. High-resolution inputs in some applications provide more feature details to be learned in hidden layers. However, high-resolution inputs might improve slightly but waste memory and computational costs since the features obtained from low-resolution images are good enough. Therefore, in the following simulation, the cropped BA images are resized to 64 × 64, 96 × 96, and 128 × 128. The accuracies are listed in the middle of Table I. The input size of 64 × 64 has performed the best in recognition. As a result of this experiment, we found that the low-resolution BA images of the cropped objects have sufficient features for CNN.

Three different CNN models are tested with the inputs obtained from the two previous simulations to obtain the best classifier. The first model consists of two convolutional layers, two pooling layers, and two fully connected layers. Then, the convolutional layers and pooling layer are increased to four in the second model. We use four convolutional layers in the third model, four pooling layers, and one fully connected layer. The performances of the three models are listed in the last three rows of Table I, and the third model has the best accuracies for pedestrians and cars. Summarizing the ablation study, the classification accuracies of the proposed algorithm are up to 94.7%, 99.3%, and 94.0% for pedestrians, cars, and street clutter, respectively.

B. Performance Evaluation

According to the previous ablation study, the proposed optimal CNN architecture consists of four convolutional layers, four pooling layers, and one fully connected layer. To further demonstrate the effectiveness of the proposed BA-CNN classifier, the accuracies of classes of our method are compared with the state-of-the-art approaches. Accuracies of three classes of PointGCN [33], PointNet [34], PointNet++ [35], 3DSSD[36], DesRNet [37], and our method are listed in Table II individually. The listed approaches are solely based on LiDAR point clouds except for the DesRNet-3 [37] since DesRNet-3 used RGB-D data.

TABLE I
THE TESTING ACCURACY OF EACH CLASS ON DIFFERENT MODELS

Factors	option	pedestrians	cars	street clutter	Accuracy
ROI Selection	**w/ min. width**	**96.7%**	**96.0%**	**93.7%**	**95.1%**
	w/ min. bounding box	92.7%	94.0%	88.0%	91.6%
Image Size	**64X64**	**92.7%**	**97.3%**	**92.7%**	**94.2%**
	96X96	90.0%	96.0%	87.3%	91.1%
	128X128	90.7%	96.0%	88.0%	91.6%
CNN Model	2 CN/ 2 FC	94.0%	96.7%	90.7%	93.8%
	4 CN/ 2 FC	86.0%	82.7%	**99.3%**	89.3%
	4 CN/ 1 FC	**94.7%**	**99.3%**	94.0%	**96.0%**

TABLE II
ACCURACY OF EACH CLASS AND AVERAGE ACCURACY ON DIFFERENT APPROACHES

Method	pedestrians	cars	street clutter	accuracy
PointGCN [33]	93.0%	99.0%	79.5%	91.0%
PointNet [34]	85.0%	98.0%	45.0%	76.0%
PointNet++[35]	85.9%	75.6%	91.4%	84.3%
3DSSD [36]	60.5%	92.6%	-	-
DesRNet [37]	95.0%	98.3%	83.0%	92.1%
DesRNet-3 [37]	**99.3%**	97.5%	**97.0%**	**97.9%**
BA-CNN	94.7%	**99.3%**	94.0%	96.0%

In Table II, DesRNet-3 has the best performance, but its model is very complicated and takes a colossal amount of memory to store the parameters. While DesRNet-3 used RGB-D data, the proposed algorithm, BA-CNN, is solely based on LiDAR point clouds transformed into 2D BA images. Because the BA image is similar to visual images, the proposed BA-CNN is faster than DesRNet-3. Despite being behind the DesRNet-3, the performance of the proposed BA-CNN is better than all listed approaches solely based on LiDAR point clouds.

V. CONCLUSION

In recent years, many approaches to point cloud processing have been proposed. Unlike 2D images without depth information, the point cloud keeps the spatial information of objects full of surface features. Therefore, 3D images have been widely used in autonomous systems and machine vision recently.

In this paper, a novel CNN-based classifier for point cloud objects is proposed to classify objects into three classes:

pedestrians, cars, and street clutters. CNN has been widely used for image recognition with great success recently. However, it is a great challenge to use CNN to recognize point cloud objects because the 3D features of the point cloud are different from 2D images, and the structure of the point cloud cannot be used in CNN directly. This paper proposes a method to transform point cloud objects into 2D images and then use CNN to identify them. The proposed algorithm consists of four steps: (1) ground point removal, (2) flood-fill clustering, (3) transforming to BA images (4) classifying by CNN. The experiment result shows that the proposed CNN-based classifier has high Accuracy. The Accuracy of car detection is up to 97.35%, 90%, and 90.4% for precision, recall, and F1-score. Pedestrian detection accuracy is 98.45%, 73%, and 88.45% for precision, recall, and F1-score. Since the shapes of street clutter are variant, the accuracy of street cutters is lower in some cases.

This paper also verifies that the bearing angle images can be used in convolutional neural networks. In future work, a CNN-based classifier using the fusion of bearing angle image and color images might significantly improve the accuracy of classification.

References

[1] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, “Rotational projection statistics for 3D local surface description and object recognition,” *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013, doi: 10.1007/s11263-013-0627-y.

[2] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino, “3D geometric scale variability in range images: Features and descriptors,” *Int. J. Comput. Vis.*, vol. 99, no. 2, pp. 232–255, 2012.

[3] A. Azim and O. Aycard, “Detection, classification and tracking of moving objects in a 3D environment,” in *Proc. IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, Jun. 2012, pp. 802–807.

[4] Y. Ye, L. Fu, and B. Li, “Object detection and tracking using multi-layer laser for autonomous urban driving,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 259–264.

[5] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, “Unsupervised feature learning for outdoor 3-D scans,” in *Proc. Australas. Conf. Robot. Autom.*, 2013, pp. 1–9.

[6] M. Lehtomäki *et. al*, “Object classification and recognition from mobile laser scanning point clouds in a road environment,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 1226–1239, Feb. 2016, doi: 10.1109/TGRS.2015.2476502.

[7] A. Börcs, B. Nagy, and C. Benedek, “Dynamic 3D environment perception and reconstruction using a mobile rotating multi-beam LiDAR sensor,” in *Handling Uncertainty Networked Structure Robot Control*. Springer, 2016, pp. 153–180. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-26327-4_7

[8] B. Yang and Z. Dong, “A shape-based segmentation method for mobile laser scanning point clouds,” *ISPRS J. Photogramm. Remote Sens.*, vol. 81, pp. 19–30, Jul. 2013.

[9] B. Yang, Z. Dong, G. Zhao, and W. Dai, “Hierarchical extraction of urban objects from mobile laser scanning data,” *ISPRS J. Photogram. Remote Sens.*, vol. 99, pp. 45–57, Jan. 2015.

[10] D. Scaramuzza, A. Harati, and R. Siegwart, “Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 4164–4169.

[11] P. M. Chu, S. Cho, Y. W. Park, and K. Cho, “Fast point cloud segmentation based on flood-fill algorithm,” in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 656–659.

[12] H.-Y. Lin and C.-H. Chang, “Depth recovery from motion blurred images,” in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Hong Kong, 2006, pp. 135–138.

[13] Z. Rozsa and T. Sziranyi, “Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted LiDAR sensor,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2708–2720, Aug. 2018, doi: 10.1109/TITS.2018.2790264.

[14] D. Zai *et al.*, “3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 802–813, Mar. 2018, doi: 10.1109/TITS.2017.2701403.

[15] F. Wu *et al.*, “Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 292–305, Feb. 2017, doi: 10.1109/TITS.2016.2565698.

[16] N. Morales, J. Toledo, L. Acosta, and J. Sanchez-Medina, “A combined voxel and particle filter-based approach for fast obstacle detection and tracking in automotive applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1824–1834, Jul. 2017.

[17] X. Xu, J. Amaro, S. Caulfield, G. Falcao, and D. Moloney, “Classify 3D voxel based point-cloud using convolutional neural network on a neural compute stick,” in *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Guilin, China, Jul. 2017, pp. 37–43.

[18] J. Huang and S. You, “Vehicle detection in urban point clouds with orthogonal-view convolutional neural network,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 2593–2597.

[19] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, “Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018.

[20] A. Börcs, B. Nagy, and C. Benedek, “Instant object detection in LiDAR point clouds,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 992–996, Jul. 2017.

[21] Z. Qiu, Y. Zhuang, F. Yan, H. Hu, and W. Wang, “RGB-DI images and full convolution neural network-based outdoor scene understanding for mobile robots,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 1, pp. 27–37, Jan. 2019.

[22] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[23] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, “PointNet: A 3D convolutional neural network for real-time object class recognition,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 1578–1584.

[24] Y. Zhuang, N. Jiang, H. Hu, and F. Yan, “3-D-laser-based scene measurement and place recognition for mobile robots in dynamic indoor environments,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 438–450, Feb. 2013, doi: 10.1109/TIM.2012.2216475.

[25] F. Cao, Y. Zhuang, H. Zhang, and W. Wang, “Robust place recognition and loop closing in laser-based slam for UGVs in urban environments,” *IEEE Sensors J.*, vol. 18, no. 10, pp. 4242–4252, May 2018, doi: 10.1109/JSEN.2018.2815956.

[26] X. Hu, X. Li, and Y. Zhang, “Fast Filtering of LiDAR point cloud in urban areas based on scan line segmentation and GPU acceleration,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 308–312, Mar. 2013.

[27] Y. Zhou *et al.*, “A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1981–1985, Nov. 2014.

[28] Y. Choe, S. Ahn, and M. Jin Chung, “Fast point cloud segmentation for an intelligent vehicle using sweeping 2D laser scanners,” in *Proc. 9th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Daejeon, South Korea, Nov. 2012, pp. 38–43.

[29] C.-C. Lin, C.-W. Lee, and L. G. Yao, “Multi-threshold based ground detection for point cloud scene,” in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Taipei, Taiwan, Sep. 2019, pp. 1–5, doi: 10.1109/AVSS.2019.8909897.

[30] K. Na, J. Byun, M. Roh, and B. Seo, “The ground segmentation of 3D LiDAR point cloud with the optimized region merging,” in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Las Vegas, NV, USA, Dec. 2013, pp. 445–450.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[32] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” in *Proc. 12th USENIX Conf. OSDI*, Nov. 2016, pp. 265–283.

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[35] Y. Zhang and M. Rabbat, "A graph-CNN for 3D point cloud classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6279–6283.

[36] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11040–11048.

[37] C.-H. Chiang, C.-H. Kuo, C.-C. Lin, and H.-T. Chiang, "3D point cloud classification for autonomous driving via dense-residual fusion network," *IEEE Access*, vol. 8, pp. 163775–163783, 2020.

Chien-Chou Lin (Member, IEEE) received the B.S. degree from the Department of Computer Science and Information Engineering, Tatung University, Taipei City, Taiwan, in 1992, and the M.S. and Ph.D. degrees from the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, in 1994 and 2004, respectively. From 2010 to 2013, he was an Assistant Professor, and since 2013, he has been an Associate Professor with the National Yunlin University of Science and Technology, Taiwan. His research interests are in robotics, point cloud processing, surface matching, and object recognition.

Chih-Hung Kuo (Member, IEEE) received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, in 1992 and 1994, respectively, and the Ph.D. degree from the Department of Electrical Engineering, University of Southern California, USA, in 2003. From 2004 to 2010, he was an Assistant Professor, and since 2010, he has been an Associate Professor with the National Cheng Kung University, Taiwan. His research interests are in digital video and audio compression and digital communication system design.

Hsin-Te Chiang received the B.S. and M.S. degrees in computer science and information engineering from the National Yunlin University of Science and Technology, Yunlin County, Taiwan, in 2016 and 2018, respectively. 2018, he Research Assistant with the National Yunlin University of Science and Technology. His research interests include robotics, path planning, and object recognition.