IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 1, 2020, accepted July 31, 2020, date of publication September 2, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021191

3D Point Cloud Classification for Autonomous Driving via Dense-Residual Fusion Network

CHUNG-HSIN CHIANG[1], CHIH-HUNG KUO[1], (Member, IEEE), CHIEN-CHOU LIN[2], (Member, IEEE), AND HSIN-TE CHIANG[2]

[1]Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan
[2]Department of CSIE, National Yunlin University of Science and Technology, Douliu 64002, Taiwan

Corresponding author: Chung-Hsin Chiang (n26061733@mail.ncku.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 108-2634-F-006-002, in part by the National Cheng Kung University and Qualcomm Collaborating Research, and in part by the Ministry of Education of the Republic of China (Taiwan) under the Teaching Innovation Project on an Issue-Oriented Approach to Narrative Competence Development.

ABSTRACT Compared with the state-of-the-art architectures, using the 3D point cloud as the input of the 2D convolutional neural network without preprocessing will restrict the feature expression of the network. To address this issue, we propose a high-precision classification network using bearing angle (BA) images, depth images, and RGB images. Due to the development of unmanned vehicles, determining how to recognize objects from the information collected by sensors is important. Our approach takes data from LiDAR and a camera and projects a 3D point cloud into 2D BA images and depth images. The RGB image captured by the camera is used to select the region of interest (ROI) corresponding to the point cloud. However, only adding input information is not enough to improve the classification ability of general convolutional neural networks. In our approach, we use a Dense-Residual Fusion Network (DRF-Net), which consists of Dense-Residual Blocks (DRBs). The Dense-Residual Fusion Network can achieve 97.92% accuracy with three input formats on a KITTI raw dataset.

INDEX TERMS Object classification, 3D point cloud, convolution neural network.

I. INTRODUCTION

Object classification is widely used in various fields, such as biomedicine, production processes, home safety, elderly care, etc. In recent years, with the development and the prospect of advanced driving assistance systems, determining how to effectively make use of the information obtained by the sensors has become an important issue. Dalal and Triggs [1] propose histograms of oriented gradients (HOG) with the linear based SVM for human detection in 2D images. Calculating the gradient (including the size and orientation) of each pixel and dividing the image into cells, the gradients in each cell are connected in a series to obtain the block-wise HOG characteristic descriptor. To acquire HOG-like features, RCNN [2] first applies CNN on object detection in 2D images. Features are easier to obtain, and the performance is improved.

The 2D images are usually taken by cameras, which are easily affected by other lighting sources. LiDAR emits a laser beam to a target and obtains the 3D point cloud through its reflection. For 3D point cloud object classification, VoxNet [3] divides a point cloud into voxels and transforms them into available features. The MVCNN [4] achieves state-of-the-art performance by rendering images from different angles of the point cloud and combining the features through view pooling. However, these data representations result in a huge number of calculations. PointNet [5] directly uses the raw data from the point cloud to perform both classification and segmentation tasks. PointNet++ [6], which is the advanced version of PointNet [5], uses a hierarchical neural network to extract local features concatenated with high level features. PointGCN [7] transforms a 3D point cloud into graphs. Using graph signal processing techniques like graph convolution and multi-resolution pooling leads to a better classification performance. Since the graph information of the point cloud plays an important role in the classification accuracy, DGCNN [8] adopts a dynamic strategy that considers both local and global features to update the graph before each edge convolution to reach state-of-the-art performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

PointHop [9] adopts k-nearest neighbors to group points in the point cloud. The points in the same group are divided into eight octants around the group center. Attributes are calculated from each octant to obtain local descriptors, of which the features are further reduced by Saab transform [10]. Furthermore, PointHop updates parameters in a feedforward fashion rather than backpropagation. By so doing, PointHop can achieve comparable classification performance while requiring much lower training complexity.

Although targets can be well classified by 3D point clouds, the amount of data in the point cloud is huge and takes a substantial amount of time to calculate. Douillard *et al.* [11] propose segmenting ground points and retaining non-ground points, which not only benefit the subsequent segmentation but also reduce the amount of data in a scene. Recent studies have shown that removing the ground points makes it easier to segment the region of interest (ROI), and projecting the ROI point clouds into 2D image makes it easier to identify the ROIs and reduces computation costs.

Combining RGB images with point clouds as input has become a trend in classification tasks and has exhibited promising performance. Recent studies demonstrate that combining additional information, such as bird's eye views or depth images, with RGB images can further improve accuracy. Börcs *et al.* [12] project the ROI point cloud into depth images which shows the outline. Lin *et al.* [13] further project ROIs into Bearing Angle (BA) images to show more details which contain the corners and the edges of ROIs. However, the textures of the obtained BA images are sometimes confusing and become counterproductive. Considering these factors, we integrate BA, Depth and RGB images by the preprocessing procedures to boost performance.

During preprocessing, the ground points are removed from the point cloud and the nonground points are then grouped and projected into the BA image and the depth image. The RGB image corresponding to the clustering result can be generated by KITTI's transformation matrix. With these various representations of ROIs as inputs, we propose a dense residual fusion network for classification.

In summary, the contributions of this article are:

1) In addition to the BA image, we add the depth projection and the RGB image corresponding to the point cloud. Through the combination of the RGB image and the information of the depth image and the bearing angle images, the input information of the neural network is full of diversity.
2) We present the Dense Residual Fusion Network (DRF-Net). The architecture uses the dense residual block, which is more conducive to transfer the information and gradients than the residual module. In addition, we also explore the neural network fusion structure, and obtain the proportion of dense residual modules required before and after feature fusion through experiments, so that the feature map can not only be fully extracted before fusion but also can be fully integrated after fusion.

The remaining of this article is composed as follows: The related works are introduced in Section II. Our approach including preprocessing procedure and the proposed network architecture is detailed in Section III. The experiment results and the error analysis are discussed in Section IV. Finally, the conclusion and future work will be provided in Section V.

II. RELATED WORK

Convolution neural networks (CNNs) have been shown to have superior performance in terms of object detection tasks. LeCun *et al.* [14] propose LeNet with 5 layers and is regarded as the pioneer of CNN. Krizhevsky *et al.* propose AlexNet [15] which applies ReLU, dropout, and max-pooling. Using ReLU as the activation function solves the vanishing gradient problem, makes the training more efficient and improves the classification accuracy. The dropout mechanism prevents the training from overfitting. Applying the pooling mechanism not only downsamples the feature map but also extracts higher level features. To achieve better performance, the structure of the CNN grows deeper and deeper. K. Simonyan and A. Zisserman propose VGG-Net [16], which has 11-19 convolution layers.

As a network becomes deeper, the degradation problem follows. In addition, it is hard to ensure that the features transmitted to the next layer represent better than those of the previous layers during the forward pass. He *et al.* [17] propose the Deep Residual Network (ResNet) which applies skip connections to achieve identity mapping. The goal of residual learning is to learn the difference between the output and the input information. Thanks to the success of the residual learning, ResNet won 1st place in the ILSVRC 2015 classification task with more than 100 layers. Based on the concept of skip connections, Huang *et al.* [18] propose dense connected network (DenseNet) which uses dense connections to allow feature reuse. Comparing the ways of information combination, ResNet uses element-wise addition, while DenseNet applies concatenation in the direction of the channel dimension. By so doing, DenseNet can increase the variation of input to enhance the representation capability and thus can achieve a lower error rate on the ImageNet dataset than ResNet. In this work, we propose the dense residual block (DRB) to integrate the advantages of feature refinement by the ResNet and feature reuse by DenseNet.

To perform tasks on object detection, R-CNN [2] first introduces region proposals within the image to detect multiple objects. Instead of classifying each region of interest (ROI), fast R-CNN [19] proposes ROI pooling, which shares the feature maps with each ROI. Faster R-CNN [20] replaces selective search [21] with the region proposal network (RPN), which aims to locate the ROIs using a CNN and thus is more efficient. YOLO [22] achieve real-time object detection with a one-stage detector that conducts object position detection and object classification in one step. Lin *et al.* [23] propose the feature pyramid network (FPN) to detect objects of different sizes. The fully convolutional network (FCN) [24]

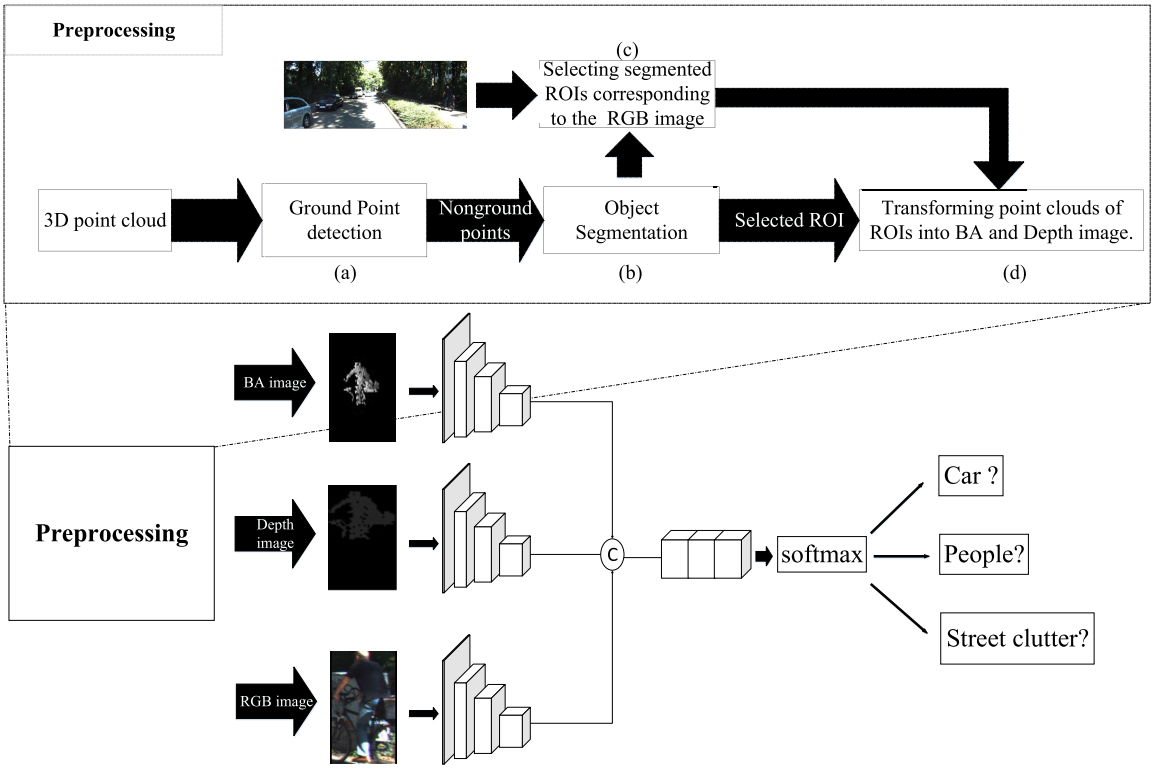

FIGURE 1. Propose classification flow.

can extract features from input images of different sizes and perform semantic segmentation.

The CNN-based methods have also been widely applied to autonomous driving with LiDAR. The MV3D [25] takes a bird's eye view, front view, and RGB images as inputs to extract feature maps which are then gathered together by a fusion network. Börcs *et al.* [12] proposed a method to detect vehicles and pedestrians by using depth images. Lin *et al.* [13] project the clustered point cloud into bearing angle images (BA images) for classification. As such, DRF-Net further takes the advantages of the above methods to make the point cloud classification more accurate.

III. APPROACH

We use a part of the KITTI raw dataset [26] as training and testing data that adopts Velodyne HDL-64E to collect point cloud information. The KITTI dataset contains various urban and suburban scenes that are very suitable for our research. Velodyne HDL-64E is a multi-beam LiDAR with 64 layers, and each layer contains 2,084 points, so there are 133376 points in a scene and 64 points in each scanline. The preprocessing pipeline shown in Fig. 1 consists of four steps: (a) remove ground points via a ground point detection algorithm; (b) produce ROIs (region of interests) from the point cloud; (c) select segmented ROIs corresponding to the RGB image, and (d) project the point cloud into bearing angle images and depth images. The BA, Depth and RGB images obtained by the preprocessing are classified by the dense residual fusion network. We will detail each step and the network architecture in the following subsections.

A. ADJUSTED THRESHOLD FOR GROUND POINT DETECTION

In a point set, the ground point accounts for 30 to 50% of the point cloud. Due to the efficiency of the ground point findings, we follow the method described in [13] as our ground point detection method. We calculate the height of the scanning point to locate the first ground point on each scanline. As shown in Fig. 3, P_i is the scanning point. With the height H of the sensor, the distance l_i between the LiDAR and the scanning point, the angle θ_i between l_i and the horizontal plane, we can obtain the height H_p of the scanning point as

$$H_p = |l_i \ \sin\theta_i + H|. \tag{1}$$

If H_p is smaller than the threshold height H_{th} which is set to 15 cm, the point P_i can be regarded as a ground point. According to the specification of Velodyne LiDAR HDL-64E, the angle θ_i lies within the range of

$$-24.8° \leq \theta_i \leq 2°. \tag{2}$$

The ground point detection method considers the height of the scanning point P_i to determine the first ground point. After detecting the initial ground point, the next ground point

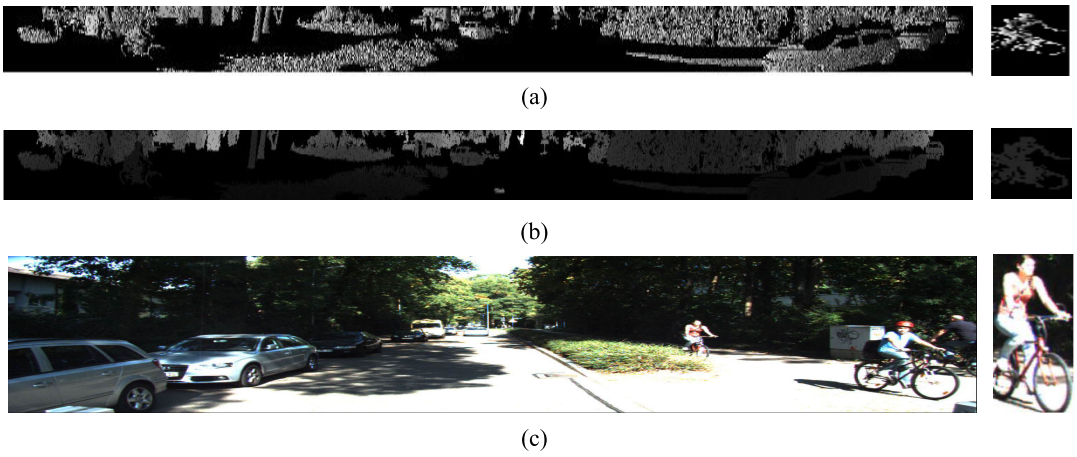

FIGURE 2. Proprojected images from non-ground points. (a) BA image, (b) Depth image, (c) RGB image.

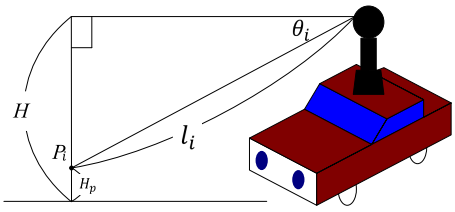

FIGURE 3. Geometric relation between a LiDAR and the detected point *Pi*.

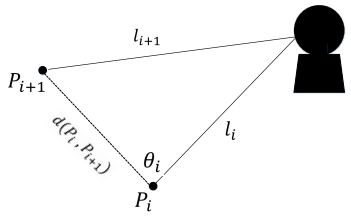

FIGURE 4. Geometric relation between a LiDAR and two consecutive points.

on the scanline is determined by the slope between the former and the next point.

Suppose a ground point is labeled P_1 (x_1, y_1, z_1), and P_2 (x_2, y_2, z_2) is the next point. The slope can be defined as

$$\theta = \frac{|y_1 - y_2|}{\sqrt{(x_1 - x_2)^2 + (z_1 - z_2)^2}}. \tag{3}$$

We observe that the points near the LiDAR are denser than those far from the LiDAR. In the cases where two consecutive points are scanned close to the LiDAR, the distance between the two points will be shorter, and when two consecutive points are scanned far from the LiDAR, the distance between them will be longer. To compensate the effect of distance on the slope, we adjust the threshold slope as

$$T_{\text{adjust}} = \begin{cases} T_0 + \alpha \cdot (d_c/d_{(1,2)})^2, & \text{if } d_c \geq d_{(1,2)} \\ T_0 - \beta \cdot (d_{(1,2)}/d_f)^2, & \text{if } d_{(1,2)} \geq d_f \\ T_0 & \text{if } d_f \geq d_{(1,2)} \geq d_c, \end{cases} \tag{4}$$

where d_c and d_f are the predetermined distance of the near area and the far area, respectively. $d_{(1,2)}$ is the distance between P_1 and P_2. T_0 is the threshold slope while the distance is between d_c and d_f. α and β are constants. If the slope θ is smaller than the threshold slope T_{adjust}, the next point would be considered to be a ground point.

B. ROI PRODUCED BY FLOOD-FILL ALGORITHM

After labeling the nonground points, they are clustered by the flood-fill algorithm [27]. The flood-fill algorithm is composed of two steps. In the first step, nonground points are clustered in each scanline. There is a threshold distance d_1 to determine whether the two consecutive nonground points belong to one cluster. If the distance between the points is smaller than d_1, they are assigned to the same cluster. In the second step, the clusters in each scanline are grouped into objects. Two threshold distances, d_h and d_v, are used to determine whether the clusters belong to the same object in the horizontal and vertical directions.

C. TRANSFORM THE 3D-POINT CLOUD INTO BEARING ANGLE AND DEPTH IMAGE AND OBTAIN RGB IMAGE

The depth image represents the proportion of the distance in gray level. The pixel value can be defined as

$$P_i^D = \frac{d_i}{d_{\text{farthest}}} \times 255, \tag{5}$$

where d_{farthest} is the distance of the farthest point in the point cloud, and d_i is the distance between the current point and the LiDAR.

According to [28], the bearing angle image (BA image) represents more details in the point cloud. Fig. 4 shows the angle θ_i between the laser beam and the line segment $d_{(P_i,Pi+1)}$ of two consecutive points P_i, P_{i+1}. To transfer the point cloud into a BA image, θ_i can be represented as

$$\theta_i = \arccos\left(\frac{l_i^2 + d_{(P_i,P_{i+1})}^2 - l_{i+1}^2}{2\, l_i\, d_{(P_i,P_{i+1})}}\right) \quad 0 \leq \theta_i \leq 180, \tag{6}$$

where l_i and l_{i+1} are the distances of P_i and P_{i+1}, respectively, measured from LiDAR. $d_{(P_i,Pi+1)}$ is the length of the line

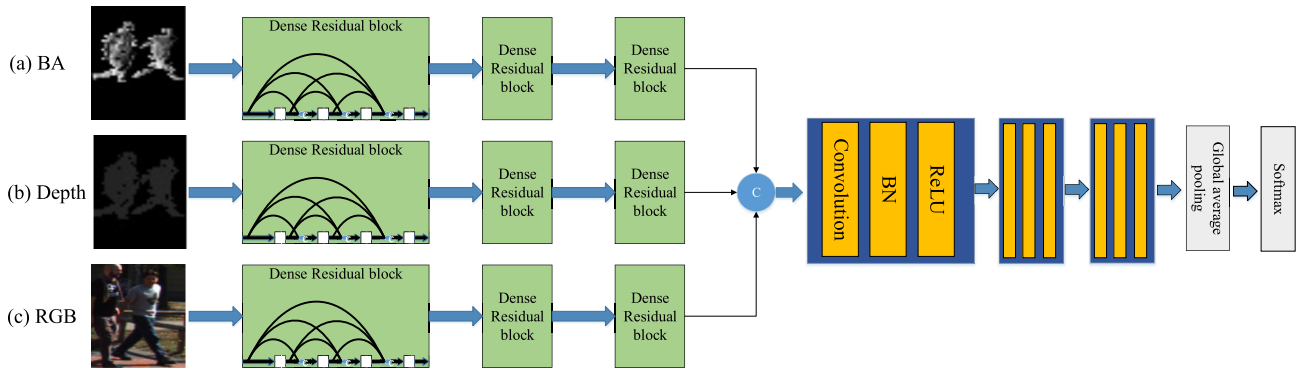

FIGURE 5. Architecture of the proposed DRF-Net. Inputs from (a) (b) (c) are BA images, depth images and RGB images, respectively.

segment connecting two consecutive points P_i and P_{i+1}. The pixel value of each point is calculated by

$$P_i^B = \frac{\theta_i}{180} \times 255. \tag{7}$$

To obtain the ROI of the RGB image, the transforming matrix provided by the KITTI dataset [26] is adopted. Coordinates of the points are projected onto the RGB image. We segment the part that corresponds to the ROI. With the RGB image, BA image, and depth image, the next stage is classifying the input images using DRF-Net. Fig. 2 shows in an exemplary frame the ROI along with the depth image, the BA image, and the RGB image. The depth image shows the contour of the frame, and the BA image contains more details that make the picture look more three-dimensional.

D. NETWORK ARCHITECTURE

The information extracted from different input formats needs to be carefully fused. Two sorts of fusion strategies, early fusion and late fusion, are considered. The early fusion strategy allows information to be fused at the front feature level. The late fusion strategy combines different local decisions from different sources to avoid the dominance by one of the input formats. We find that the early fusion performs better than the late fusion in extensive experiments, so we adopt the early fusion in our DRF-Net.

The DRF-Net architecture is shown in Fig. 5. The network consists of three dense-residual blocks to extract features from each input source. The features are concatenated and processed by three convolution layers to learn higher level representations.

Fig. 6 shows the structure of the dense residual block (DRB). Each dense-residual block is composed of three residual blocks [17] followed by one max-pooling layer and one convolution layer with 1×1 kernel size which is used to reduce the dimensionality. Batch normalization and ReLU are attached after each convolution layer. Batch normalization (BN) was proposed to solve the problem of the internal covariate shift in [29]. BN reduces the sensitivity of the model

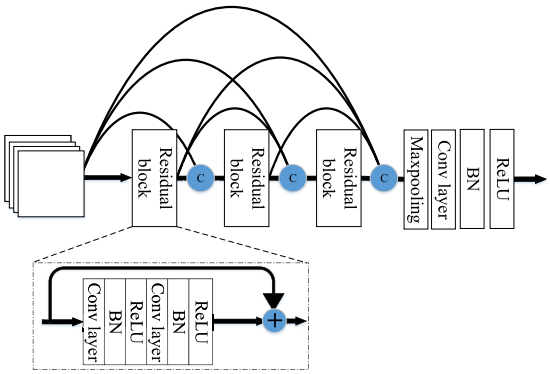

FIGURE 6. The structure of the dense residual block (DRB).

to network parameters, makes the network learning more stable, and the training speed is faster.

The residual block includes two convolution layers with a shortcut. The output of ith residual block can be defined as:

$$R_i = \sigma(\mathrm{BN}(W_{i,2}(\sigma(\mathrm{BN}(W_{i,1}x_i))))) + x_i, \tag{8}$$

where x_i is the input of the residual block and $W_{i,c}$ is the cth convolution layer in the ith residual block. BN denotes the operation of batch normalization and σ is the ReLU activation function.

When the network gets deeper, it will make the low-level information disappear after multiple stacked layers. In order to reuse feature maps and increase information flow, we add dense connections. The output of the l th Dense Residual Block DRB_l can be shown as:

$$DRB_l = \sigma(\mathrm{BN}(W_{l,(1\times1)}M_l)), \tag{9}$$

where $M_l = \mathrm{MaxPool}(x_l \oplus R_{l,i} \oplus R_{l,i+1} \oplus R_{l,i+2})$, M_l is regarded as the output of the maxpooling layer and x_l is the input feature map of lth DRB. The symbol $\oplus$ denotes the concatenation operation. $R_{l,i}$ is the output of ith residual block and $W_{l,(1\times1)}$ is the 1×1 convolution layer in the lth DRB.

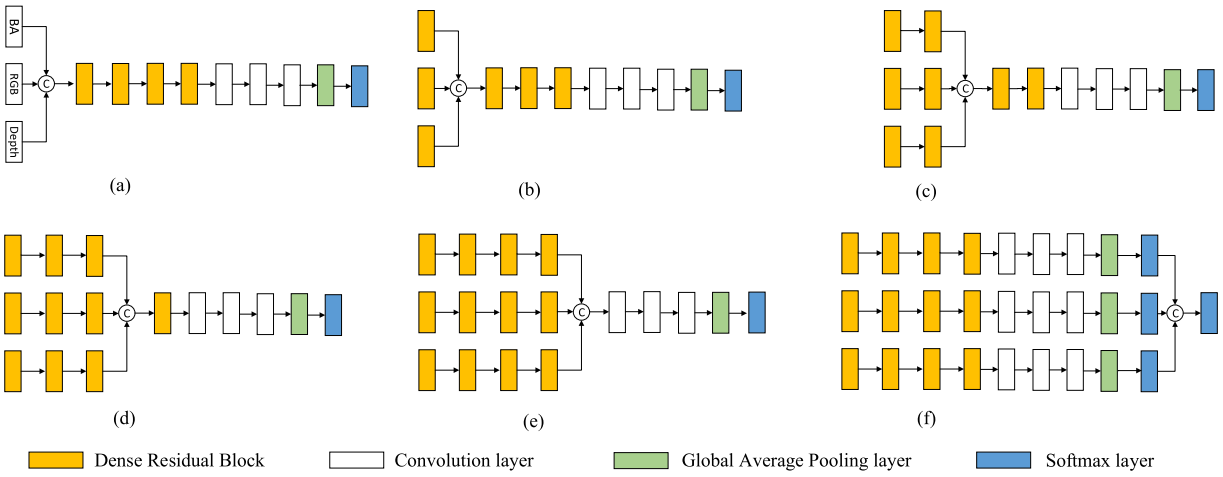

FIGURE 7. Various network architectures with different DRB layers before feature fusion. (a) Early fusion without DRB layer, (b) 1 DRB layer, (c) 2 DRB layer, (d) 3 DRB layer, (e) 4 DRB layer, and (f) Late Fusion.

E. LOSS FUNCTION

Generally, classification networks use cross entropy as a loss function, regardless of whether the sample is difficult or easy. However, most of the samples can be classified easily. Thus, well-classified examples will comprise most of the loss during training. Using the cross entropy as cost function does not well handle this situation. To reduce the loss for easy examples and focus more on hard examples, we use the focal loss proposed in [30] instead of cross entropy. The focal loss can be written as

$$\mathrm{FL}(P_t) = -(1 - P_t)^{\gamma} \ln(P_t), \tag{10}$$

where P_t denotes the probability of the final prediction. γ is the parameter which is set to downweight the well-classified examples. We set $\gamma = 2$ in our experiment.

IV. EXPERIMENTS

Five scenes from the KITTI dataset [26] are adopted in our experiments. The raw data from Residential 2011_09_26_drive_0035 and Campus 2011_09_28_drive_0021 comprise the training set. The testing set contains the raw data for Residential 2011_09_26_drive_0020, 2011_09_30_drive_0027 and Campus 2011_09_28_drive_0039. The input images are divided into three categories: pedestrians, cars, and street clutter. In our dataset, we classify cyclists as pedestrians. There are totally 2,000 images in the training set and 1,200 images in the testing set (400 images in each category). The network is trained with AdamOptimizer using Tensorflow, wherein the parameters β_1 and β_2 are set to 0.9 and 0.99, respectively. The learning rate is set to be 0.0005. We run the proposed RF-Net on GTX 1080 Ti GPU with a batch size of 16 for 200 epochs. The input images are resized into a fixed resolution of 96×96.

A. ABLATION STUDIES

In Table 1, we investigate the impact of various input combinations on the output accuracy. When taking the RGB image as the single input, the performance of our proposed network is better than the other alternatives. The depth image shows the texture of the point cloud in 2D. The BA image enhances the details of outlines and corners. The RGB information makes it easier to recognize the object in each ROI. The fusion of the BA image with the RGB image improves the accuracy from 90.25% to 96.50%. The fusion of the depth image with the BA image makes the extracted features more robust, and hence improves the accuracy from 87.17% to 92.08%. The fusion of the depth image with the RGB image improves the accuracy from 87.17% to 96.80%. Fusing all three types of features leads to the best performance of accuracy 97.75%. In short, entering three types of images simultaneously for classification can indeed improve overall detection accuracy.

To compare the DRB with the residual block, Table 2 shows the results of the 2-input fusion networks that applying the residual block [4] and the dense residual block, respectively. The reason we choose only 2 input sources is to save training time. It can be noticed that the model using the dense residual blocks achieve better average accuracy than that using residual blocks by 1.4% to 6.4%. Dense connections not only transmit more information flow within a block, but also efficiently prevent both vanishing and explosive gradients.

After concatenating feature maps extracted from different input sources, we need to combine them for the feature integration. As Table 3 shows, adding a dense residual block after feature fusion improves the accuracy from 97.75% to 97.92%. We accordingly take our DRF-Net with an extra DRB as the baseline model in the following experiments.

Fig. 7 illustrates different combinations of DRB numbers used before and after fusion that may affect the accuracy. In Table 4, the accuracy improves as the number of the DRBs

TABLE 1. Average testing accuracies of the dense residual fusion netowrk for different input combinations. (single: BA/Depth/RGB image, 2-input: BA + Depth/BA + RGB/Depth + RGB, 3-input: BA + Depth + RGB).

	single-input	single-input	single-input	2-input	2-input	2-input	3-input
BA image	√			√		√	√
Depth image		√		√	√		√
RGB image			√		√	√	√
Average Accuracy (%)	90.25	87.17	94.00	92.08	96.80	96.50	**97.75**

TABLE 2. Average testing accuracies of 2-input fusion networks for using residual blocks and dense residual blocks.

Accuracy (%)	BA+RGB	BA+Depth	Depth+RGB
Residual block	94.00	90.60	90.40
Dense Residual block	**96.50**	**92.08**	**96.80**

TABLE 3. Average testing accuracies for DRB after fusion, where CA denotes channel attention.

Model	Average Accuracy(%)
DRF-Net	97.75
DRF-Net(Add a DRB)	**97.92**
DRF-Net(Add a DRB + CA)	97.83

TABLE 4. Average accuracies various fusion models.

Model	(a)	(b)	(c)	(d)	(e)	(f)
Accuracy (%)	89.25	95.08	95.50	**97.92**	97.08	95.00

TABLE 5. Accuracies for two loss functions.

Loss function	Car	People	Street	Average
Cross Entropy	**98.70 %**	96.00 %	96.25 %	97.00 %
Focal Loss	97.50 %	**99.25 %**	**97.00 %**	**97.92 %**

TABLE 6. Average testing accuracies for different models, the symbol "∗" denotes DRF-Net with 3-input.

Model	car	pedestrians	street clutter	Accuracy (%)
PointGCN [7]	99.00	93.00	79.50	91.00
PointNet [5]	98.00	85.00	45.00	76.00
PointHop [9]	**99.50**	96.50	77.25	91.33
DGCNN [8]	91.20	89.33	82.50	87.66
PointNet++ [6]	75.60	85.90	91.40	84.30
Chiang et al [13]	62.00	91.70	62.90	72.20
Proposed	98.25	95.00	83.00	92.08
Proposed*	97.50	**99.25**	**97.00**	**97.92**

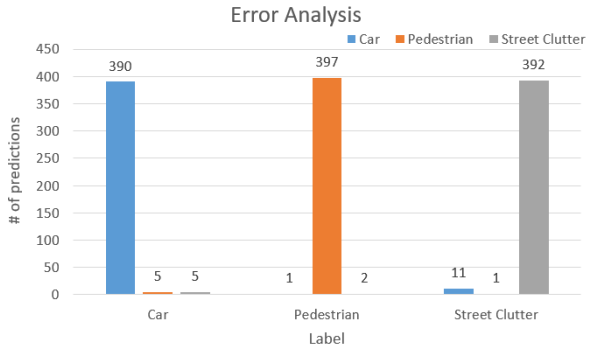

FIGURE 8. Error analysis of the model with the best performance.

before fusion increase from (a) to (d). It can be observed that the more DRBs increase before fusion, the better feature is extracted. In (e), the decreased accuracy shows the importance of feature integration after fusion. In (f), a decision fusion structure is used, which combines the predictions from each input obtain worse accuracy than (d) and (e). As a result, fusing the information in the feature level is more appropriate than fusing in the decision level. We find that model (d) has the best accuracy of 97.92% and thus we choose (d) as our final model.

Loss function plays an important role while training. Compared to the cross entropy, the focal loss pays more attention on hard examples. Table 5 compares the effects of the focal loss and cross entropy used to train our best model (Fig. 7(d)). Although the focal loss drops the precision for classifying cars by 1.2%, it improves the accuracy for pedestrians and street clutter by approximately 3.25% and 0.75%, respectively. In general, the focal loss appears to be more suitable than the cross entropy to train the proposed model.

We also investigate the impact of attention mechanism. We adopt the attention module [31] propose by Hu et al to learn which channels in the DRBs are worth putting more weights. Table 3 shows that the scheme with channel attention (CA) mechanism decreases the accuracy by 0.09%. The reason for the accuracy drop may be that the channel weights computed by the Sigmoid function are always smaller than 1, and consecutive multiplications with these weights will make the feature values become smaller and smaller. Consequently, the global average pooling may fail to extract features as the basis, and hence we do not adopt attention mechanism in the proposed DRF-Net.

With the widespread use of LiDAR, many new architectures are proposed to perform classification for point clouds. To be compared, point sets segmented by our preprocessing procedure are input to these models. Table 6 lists the accuracies of models proposed by PointGCN [7], PointNet [5], PointNet++ [6], PointHop [9], DGCNN [8], and Lin *et al.* [13]. The model by Lin *et al.* [13] projects the point clouds into BA images, while those of DGCNN and PointGCN convert the point clouds into graph signals. The other models' input point clouds to the neural networks without pre-processing. For a fair comparison, we also listed the results of our DRF-Net without the RGB input image in Table 6. Compared with other models, our model keeps stable classification accuracy for all three classes and achieves an average accuracy of 92.08%. The inclusion of RGB information further enhances the accuracy of street clutter and pedestrians, and reaches to the best accuracy 97.92%.

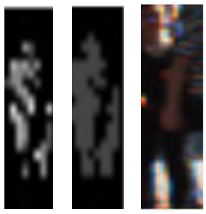

FIGURE 9. A pedestrian in different input sources. (a) BA image, (b) depth image, (c) RGB image.

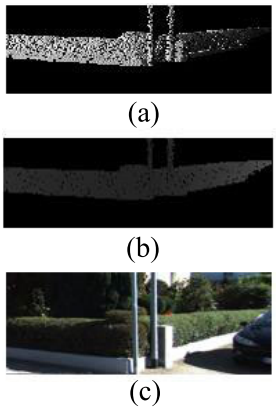

FIGURE 10. Street clutter in different views. (a) BA image, (b) depth image, (c) RGB image.

B. ERROR ANALYSIS

We analyze the classification error for our proposed 3-input model by tracking the distributions of prediction results in each category in Fig 8. Our proposed model performs well in identifying pedestrians. Most of the mispredictions occur when pedestrians are classified as street clutter, accounting for 2 mispredictions (0.5%). As shown in the RGB image of Fig 9(c), the pedestrian overlaps with the other person's hand. This misclassification comes from the reduced resolution due to the long distance between LiDAR and object. The error rates of regarding cars as pedestrians and street clutter are similar. Most of the false predictions take place in the scenes which consider street clutter as cars, accounting for 11 mispredictions (2.75%). Fig 10 shows an example of street clutter, which is misclassified as a car due to the overlapping of different objects. In order to improve accuracy, the pre-processing may need to include more semantic information from RGB images to help perform segmentation of ROIs.

V. CONCLUSION

In this article, we propose a framework that projects a 3D point cloud into 2D images as input to the DRF-Net. The DRF-Net leverages dense residual blocks to extract features from multiple input sources, which in turn leads to better classification performance. We also explore the fusion structures to further improve the accuracy of classification. Compared to other classification models, the proposed DRF-Net achieves better accuracy by transforming the point cloud into BA and depth images. For future work, we need to tackle similar issues faced by the R-CNN. For example, there are too many ROI selection processes that may incur excessive computations. Since all selected ROIs have to perform classification by a neural network, the system memory may run out quickly. Inspired by the Faster R-CNN, we would attempt to combine RPN with ROI to implement a more practical classification scheme.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.
[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[3] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
[4] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
[6] 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.
[7] 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.
[8] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.
[9] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C.-J. Kuo, "PointHop: An explainable machine learning method for point cloud classification," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1744–1755, Jul. 2020.
[10] C.-C.-J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 346–359, Apr. 2019.
[11] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LiDAR point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 2798–2805.
[12] A. Borcs, 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.
[13] H. T. Chiang, "3D Point Cloud Classification using Convolutional Neural Network," M.S. thesis, Nat. Yunlin Univ. Sci. Technol., Douliu, Taiwan, Jun. 2018.
[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[21] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[25] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[27] 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.
[28] 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.
[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

CHUNG-HSIN CHIANG was born in Nantou, Taiwan, in 1994. He received the B.S. degree in engineering science and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan. His research interests include deep learning, image processing, and so on.

CHIH-HUNG KUO (Member, IEEE) received the B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, in 1992 and 1994, respectively, and the Ph.D. degree from the University of Southern California (USC), Los Angeles, CA, USA, in 2003, all in electrical engineering.

He was with the Computer and Communications Research Laboratories/Industrial Technology Research Institute (CCL/ITRI), Taiwan, as a DSP Design Engineer, from 1996 to 1998. Since March 2004, he has been a Senior Engineer at Winbond Electronics, Taiwan. In August 2004, he joined the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, as an Assistant Professor. He has been an Associate Professor, since February 2010. His current research interests include system-level designs for video processing and multimedia communications.

CHIEN-CHOU LIN (Member, IEEE) received the M.S. and Ph.D. degrees from National Chiao Tung University, 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 include robotics, point cloud processing, surface matching, and object recognition.

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, Douliu, Taiwan, in 2016 and 2018, respectively. Since 2018, he has been a Research Assistant with the National Yunlin University of Science and Technology. His research interests include robotics, path planning, and object recognition.