applied sciences

MDPI

*Article*

# PIFNet: 3D Object Detection Using Joint Image and Point Cloud Features for Autonomous Driving

**Wenqi Zheng [1], Han Xie [1], Yunfan Chen [2], Jeongjin Roh [1] and Hyunchul Shin [1,*]**

[1] Department of Electrical Engineering, Hanyang University, Ansan 15588, Korea; zhengwenqi@hanyang.ac.kr (W.Z.); xiehan@hanyang.ac.kr (H.X.); jroh@hanyang.ac.kr (J.R.)
[2] School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China; yfchen@hbut.edu.cn
* Correspondence: shin@hanyang.ac.kr; Tel.: +82-10-3775-5176

**Abstract:** Owing to its wide range of applications, 3D object detection has attracted increasing attention in computer vision tasks. Most existing 3D object detection methods are based on Lidar point cloud data. However, these methods have some limitations in localization consistency and classification confidence, due to the irregularity and sparsity of Light Detection and Ranging (LiDAR) point cloud data. Inspired by the complementary characteristics of Lidar and camera sensors, we propose a new end-to-end learnable framework named Point-Image Fusion Network (PIFNet) to integrate the LiDAR point cloud and camera images. To resolve the problem of inconsistency in the localization and classification, we designed an Encoder-Decoder Fusion (EDF) module to extract the image features effectively, while maintaining the fine-grained localization information of objects. Furthermore, a new effective fusion module is proposed to integrate the color and texture features from images and the depth information from the point cloud. This module can enhance the irregularity and sparsity problem of the point cloud features by capitalizing the fine-grained information from camera images. In PIFNet, each intermediate feature map is fed into the fusion module to be integrated with its corresponding point-wise features. Furthermore, point-wise features are used instead of voxel-wise features to reduce information loss. Extensive experiments using the KITTI dataset demonstrate the superiority of PIFNet over other state-of-the-art methods. Compared with several state-of-the-art methods, our approach outperformed by 1.97% in mean Average Precision (mAP) and by 2.86% in Average Precision (AP) for the hard cases on the KITTI 3D object detection benchmark.

**Keywords:** 3D object detection; lidar point cloud; camera images; object detection



**Citation:** Zheng, W.; Xie, H.; Chen, Y.; Roh, J.; Shin, H. PIFNet: 3D Object Detection Using Joint Image and Point Cloud Features for Autonomous Driving. *Appl. Sci.* **2022**, *12*, 3686. https://doi.org/10.3390/app12073686

Academic Editor: Antonio Fernández-Caballero

Received: 4 March 2022
Accepted: 4 April 2022
Published: 6 April 2022



## 1. Introduction

Object detection plays an important role in an Advanced Driver Assistance System (ADAS). Since the accuracy of the object detection in an ADAS is essential for the safety of drivers and pedestrians, it is considered one of the most challenging computer vision problems. The traditional object detection based on images is to transform the detection task into a classification task by using a sliding window. These classifiers usually use manual features and classifiers like Support Vector Machines (SVMs) [1,2]. The main weakness of this method is that the amount of computation is very large because the classifier must scan each position in the space. Light Detection and Ranging (LiDAR) sensors are important for ADAS to perceive the environment and obstacles. Therefore, it is crucial to effectively process irregular and uneven LiDAR point clouds. The classical method [3] to process LiDAR data in 3D detection task achieved good performance. For example, Gao et al. [3] proposed a dynamic clustering algorithm that applied an elliptical function to adjust each position of key points. Convolution Neural Networks (CNNs) have improved the performance of object detection because CNNs can extract useful features

from 2D images, learn features, and classify objects. Many researchers have proposed various neural network architectures for 2D object detection, and these architectures can be extended for 3D object detection tasks. In [4], the authors proposed to generate specific object proposals instead of classification by using deep learning. With the emergence of more and more 3D object detection datasets and sensors, developing new 3D object detection algorithms attracts more interest. Recently the majority of state-of-the-art and popular 3D object detection methods utilize deep learning technologies.

Most existing 3D detection methods based on deep learning can be classified into three categories in terms of representations, i.e., the point cloud-based methods [5–19], camera image-based methods [20–23], and the point cloud and image fusion-based methods [24–32]. In general, it is not easy to achieve good performance in 3D object detection by using only camera sensors because there is no depth information in RGB images. Thus, LiDAR sensors and RGB-D camera sensors are extensively used to support the depth information for 3D object detection.

In the last decade, significant progresses have been reported in 3D object detection using RGB images and LiDAR point clouds. Rich semantic information in color and texture can be provided by camera RGB images, whereas depth information and geometric structure features can be provided by LiDAR point clouds. Therefore, effective fusion of camera images and LiDAR point clouds appears to be extremely beneficial for understanding 3D objects and 3D scene construction. LiDAR sensors are commonly used in various self-driving vehicles and robot applications to capture the surrounding 3D scene information. Although point clouds provide depth information for environmental perception and 3D structure understanding, LiDAR points are usually sparse and unordered. We designed an effective fusion module to combine data from different sensors targeting a more accurate 3D object detection.

In this paper, we propose a new 3D object detection framework, Point-Image Fusion network (PIFNet), which fuses camera image features and LiDAR point cloud features. In fact, the problem of fusing camera and LiDAR sensors is challenging as the features obtained from camera images and LiDAR point clouds are represented from different points of views, i.e., camera-view and 3D world view. The information fusion is achieved using two object detection streams. In one stream, we use PointNet++ [11] to generate the point-wise features. In the other stream, we aim to extract the image features and generate the strong joint Camera-LiDAR features.

The main contributions of our work can be summarized as follows.

- We designed a new 3D object detection method, PIFNet, by using the fusion of LiDAR and camera sensors. To compensate for the sparsity and the irregularity of point clouds for far objects, camera images can provide rich fine-grained information. The point cloud features and image features are fused to take advantage of the complementary characteristics of these two sensors.
- We developed an Encoder-Decoder Fusion (EDF) network to effectively extract image features. To obtain the multi-scale feature maps, we used multiple-level feature maps. The low-level feature maps have high resolution but poor semantic information whereas the high-level feature maps have low resolution but dense semantic information. By using this EDF module, we can obtain accurate localization information from low-level feature maps and useful semantic classification information from high-level feature maps.
- A fusion module is proposed to integrate the color and texture features from images and the depth information from point clouds. Each intermediate feature map is fed into the fusion module to integrate with its corresponding point-wise features. This module can cope with the irregularity and sparsity of the point cloud features by capitalizing the fine-grained information extracted from images by the EDF module.
- The experiments using our method and several other state-of-the-art methods show that our proposed neural network architecture can produce the best results on average.

In particular, on the KITTI dataset, PIFNet achieved 85.16% mAP (the best results) in 3D object detection and 90.96% mAP (second best results) in BEV detection.

The rest of this paper is organized as follows. In Section 2, the related works of 3D object detection are briefly reviewed. Our proposed PIFNet is presented in Section 3. The experimental results are described in Section 4. Finally, the conclusions are given in Section 5.

## 2. Related Works

In this section, we review several 3D object detection techniques.

### 2.1. 3D Object Detection Based on Camera Images

Recently, plenty of 3D detectors have been proposed by using camera images, such as monocular [20,21] and stereo images [22,23]. Ma et al. [20] exploited 3D features in the reconstructed 3D space and generated class-specific 3D object proposals by using monocular images. Xu et al. [21] presented a framework to generate 2D region proposals and predicted 3D locations and orientations by using the disparity map. However, it is hard to obtain accurate 3D bounding boxes by using the methods based on the monocular camera images because of the lack of depth information. Some methods [22,23] use stereo images to add the depth map for 3D object detection. However, the depth information provided by stereo images is limited and not accurate when compared to the information by LiDAR point clouds.

### 2.2. 3D Object Detection Based on LiDAR

Many LiDAR-based methods are proposed in recent years. Barrera et al. [19] extended BirdNet to BirdNet+ to perform regression of height and vertical location of the center point of a 3D bounding box. Most LiDAR-based methods generate 3D proposals based on voxel-wise features [8,16–18] or point-wise features [12–15]. Voxel-based point encoding methods use 3D regular voxels against the disorder of point clouds, but they often miss precise point positions. Yan et al. [16] developed a Sparsely Embedded CONvolution Detection (SECOND) technique to maximize the use of 3D information. Shi et al. [18] presented a part-aware and aggregation neural network to predict high quality 3D proposals. Alex et al. [17] developed a PointPillars network to convert the point cloud into a stacked pillar tensor and pillar index tensor. Deng et al. [8] voxelized to extract voxel-wise features from point cloud, converted the 3D volumes into Bird Eye View (BEV) plane to obtain region proposals, and employed voxel Region of Interest (RoI) pooling to refine the bounding boxes. The LiDAR-based 3D object detectors directly utilize raw points, but the computation cost is large. Charles R. et al. [10] introduced a pioneer, PointNet, that extracts point-wise features by Multi-Layer Perceptron (MLP) from raw point clouds without voxelization and developed hierarchical PointNet++ [11] that applied PointNet [10] recursively on raw point set. Shi et al. [14] employed PointNet [10] to predict bottom-up 3D proposals and used RoI pooling to refine them. Sparse To Dense (STD) network [15] used PointNet [10] to yield the global features representing the geometric structure of the entire point set.

### 2.3. 3D Object Detection Based on the Fusion of Camera Images and LiDAR Point Clouds

To exploit the advantages of the camera and LiDAR sensors, various camera and LiDAR fusion methods [25–27,30–32] have been proposed for 3D object detection. The approaches proposed in [13] detected the objects in two sequential steps, where the region proposals were generated in the first step based on the camera image, and then the LiDAR points in the region of interest were processed in the second step to detect the objects. However, the performance of these methods is limited by the accuracy of the camera-based detectors. In [31], Shi et al. proposed a two-stage Multi-View 3D object detector (MV3D), where LiDAR point clouds are projected to Bird Eye View (BEV), accurate multi-view 3D proposals are generated, and the region-wise features are extracted by using a multi-view fusion network. Jason et al. [26] proposed Aggregate View Object Detection (AVOD)

to integrate LiDAR BEV and camera front-view features for Region Proposal Networks (RPNs) and a second stage detection network. However, projection methods [26,33] may miss spatial structure information. Huang et al. [27] proposed a two-stream network architecture EPNet and designed a LiDAR-Image Fusion module to generate the point-wise correspondence. Liang et al. [24] presented a Multi-task Multi-sensor Fusion (MMF) detector, where 4 tasks including 2D and 3D object detection, ground estimation, and depth completion. Charles R. et al. developed Frustum Point-Net [30] that first used a mature 2D detector to generate candidate boxes on images and then extracted the point clouds from the 3D bounding frustums for box refinement. Meyer et al. [32] presented an end-to-end detector without requiring 2D image labels to associate LiDAR points with image pixels. Most of these methods [24,26,31,32] project the objects detected in 2D images to 3D space and cannot fully utilize the semantic features from images. Our method can utilize more detailed localization information from camera sensors.

## 3. Proposed Point-Image Fusion Network (PIFNet)

In this paper, we propose the Point-Image Fusion network (PIFNet), a two-stream 3D detection framework aiming to exploit the point-based and image-based features more effectively. We directly extract the point-based features from the point clouds and use the Encoder-Decoder Fusion module to extract features from images.

### *3.1. Two Stream Network Architecture*

Our two-stream network consists of a geometric stream and a semantic stream. As shown in Figure 1, the geometric stream and the semantic stream produce the point-based features and RGB image features, respectively. We developed a fusion module to effectively fuse the point-based features and image features. A multi-level fusion module is used to enhance the discriminative feature representations in multiple scales.

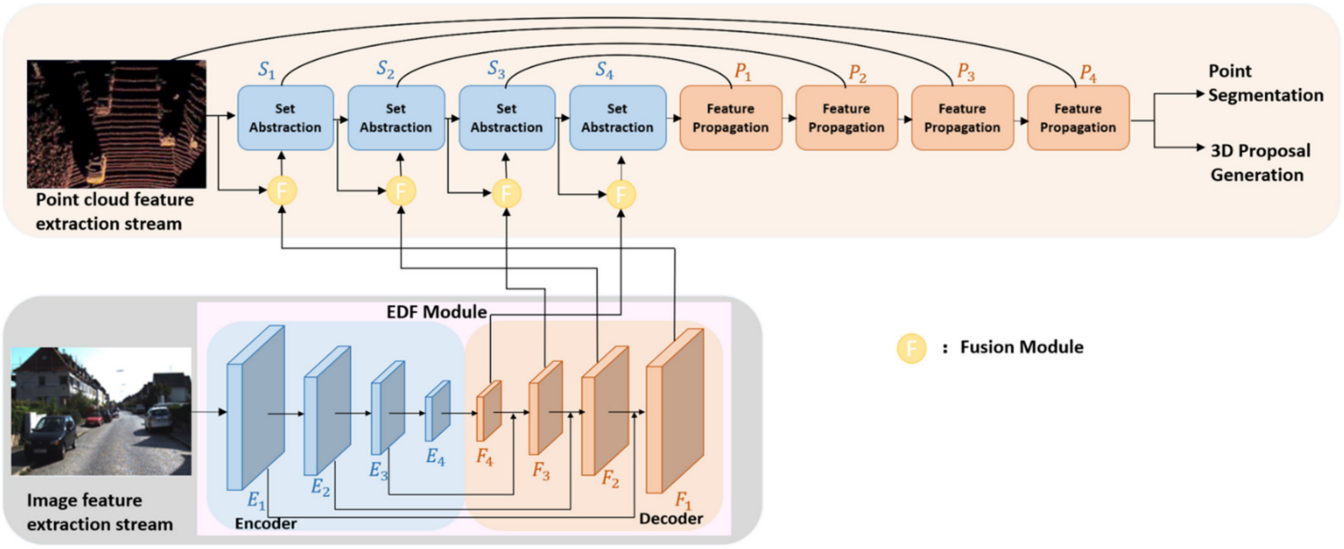


**Figure 1.** The overall architecture of PIFNet.

As illustrated in Figure 1, PIFNet consists of two parts, a point cloud feature extraction stream and an image feature extraction stream. In the image feature extraction stream, camera images are fed into the Encoder-Decoder Fusion (EDF) module. In this module, the image is passed through convolution and deconvolution layers, and the four feature maps obtained from the decoder will be integrated with a corresponding point-wise feature by the fusion module. In the point cloud feature extraction stream, the LiDAR point cloud is processed by a series of Set Abstraction (SA) modules and Feature Propagation (FP) modules. Each output obtained by the SA module will be fused with the corresponding

camera image features. Considering the tradeoff between the performance and computation time, we use 4 pairs of SA layers and FP layers to extract point-wise features. The fused features are used to generate accurate 3D proposals.

#### 3.1.1. Image Feature Extraction Stream

The camera RGB images are processed by an Encoder-Decoder Fusion (EDF) module as shown in Figure 2. The EDF module consists of a set of convolution and deconvolution layers. The encoder module uses four convolution blocks to reduce the resolution of the feature maps. Then, the feature maps in the decoder are up-scaled via deconvolution layers and then combined with the corresponding feature maps of the same resolution in the encoder through element-wise accumulation.

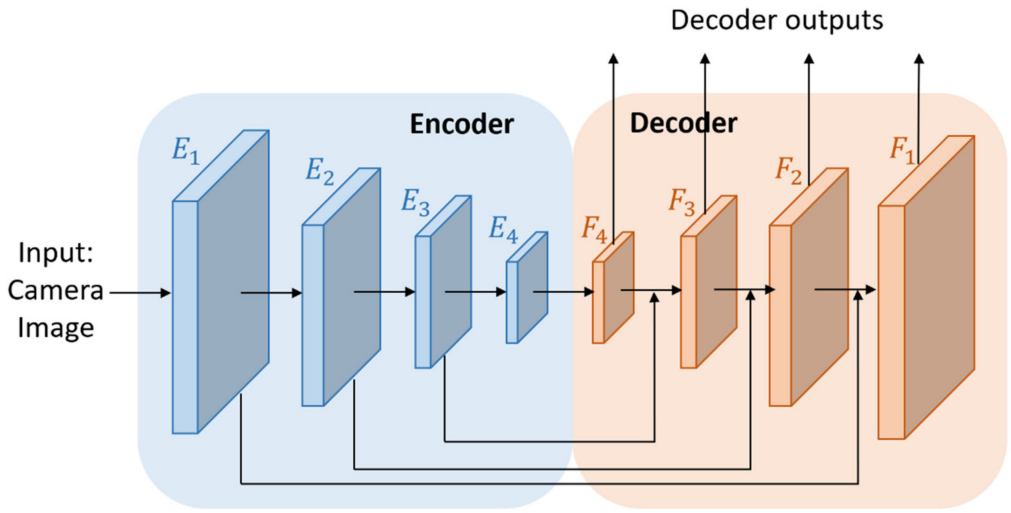


**Figure 2.** Encoder-Decoder Fusion (EDF) module architecture.

In the encoder module, the low-level feature maps have high resolution but poor semantic features. On the contrary, the high-level feature maps have rich sematic features but low resolution. The EDF module operates on the camera images and effectively extracts the semantic features and fuses the low-level feature maps and high-level feature maps to achieve good performance.

The encoder module consists of a series of convolutional blocks. Specifically, each convolutional block consists of two $3 \times 3$ convolution layers and is followed by a batch normalization layer and a ReLU activation function. Each convolution block produces a feature map $E_m$ $(m = 1, \ldots, 4)$. The decoder module consists of a series of deconvolution layers with successively increasing resolution. Each deconvolution module produces a feature map $F_n$ $(n = 1, \ldots, 4)$. $E_4 = F_4$ and $F_n$ have an increased resolution than that of the previous feature maps $F_{n+1}$, for $n$ = 1, 2, 3. We exploit the abundant texture provided by camera images by using the EDF module and fed the four feature maps into the fusion module to associate with point-wise features.

#### 3.1.2. Point Cloud Feature Extraction Stream

The point cloud feature extraction stream takes LiDAR point clouds as input and generates the 3D proposals. Considering the tradeoff between the performance and computation time, the geometric stream uses four pairs of Set Abstraction (SA) layers [11] and Feature Propagation (FP) layers [11], for point-wise feature extraction. For the convenience of description, the outputs of SA and FP layers are denoted as $S_i$ and $P_i$ (I = 1,2,3,4), respectively. As shown in Figure 1, the point feature map Si is combined with the semantic image feature $F_i$ in our fusion module. In addition, the point feature $P_4$ is further enriched by the multi-scale image feature functional units to obtain a compact and discriminative

feature representation, which is then fed to the detection modules for foreground point segmentation and 3D proposal generation.

### *3.2. Fusion Module*

To merge the two streams, we developed an image-guided feature fusion module, as shown in Figure 3. The fusion module has two inputs: lidar features and image features. The fusion module consists of a convolution block for LiDAR features, a deconvolution block for image features, and an element-wise fusion block.

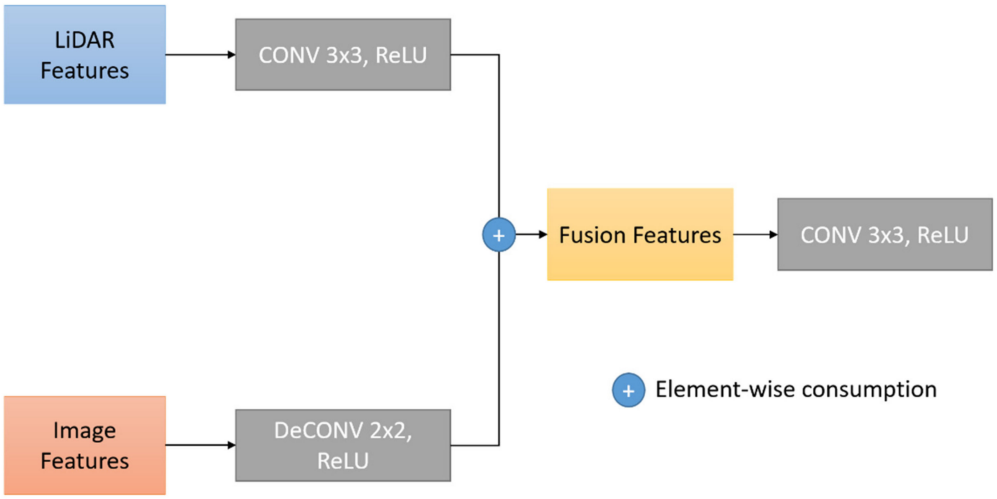


**Figure 3.** The fusion module architecture.

We can easily associate image features with LiDAR features by using the fusion module. Specifically, we use a $3 \times 3$ CONV layer and a ReLU activation function as a point-wise feature processor. The point-wise feature processor takes LiDAR point clouds and the camera images under the same resolutions and produces the point-wise correspondence between them. A $2 \times 2$ deconvolution layer and a ReLU activation function are used in the image feature processor. The point-wise features and the image features are fused by element-wise consumption to obtain fusion features. Finally, we apply a $3 \times 3$ convolution and a ReLU activation function to resize the feature.

### *3.3. Training Loss*

A multi-task loss function is used to train our neural network, PIFNet. Following EPNet [27], the total loss function is defined as the summation of training objectives for the two-stream Region Proposal Network (RPN) and the refinement network. The loss function can be formulated as follows,

$$L_{total} = L_{rpn}r + L_{rcnn} \tag{1}$$

where $L_{rpn}$ is the loss of RPN network, and $L_{rcnn}$ is the loss of refinement network.

Both of them adopt a similar optimization goal, to minimize classification loss, regression loss and Consistency Enforcing (CE) loss, i.e.,

$$L_{rpn} = L_{cls} + L_{reg} + \lambda L_{ce} \tag{2}$$

where $L_{cls}$ is the classification loss, $L_{reg}$ is the regression loss, and $L_{ce}$ is the CE loss.

The focal loss is used as the classification loss to balance the positive and negative samples. We use the same parameters, $\alpha = 0.25$ and $\gamma = 2.0$ as in [27] in the following equation. For a bounding box, the network needs to regress its center point (x, y, z), size (l, h, w), and orientation $\theta$. We directly calculate the Y-axis offset to the ground truth and the size of the bounding box (h, w, l) with a smooth L1 loss [33]. As for the X-axis, the Z-axis,

and the orientation θ, we adopt the bin-based regression loss. The loss functions can be written as follows,

$$L_{cls} = -\alpha(1 - c_t)^{\gamma} log c_t \tag{3}$$

$$L_{total} = L_{rpn} + L_{rcnn}\ L_{reg} = \sum_{u \in x,z,\vartheta} E\left(b_u, \widehat{b_u}\right) + \sum_{u \in x,y,z,h,w,l,\vartheta} S(r_u, \widehat{r_u}) \tag{4}$$

where $E$ and $S$ denote the cross-entropy loss and the smooth L1 loss, respectively. The parameter $c_t$ is the probability of the point, in consideration, belonging to the ground truth category. The parameters $\widehat{b_u}$ and $\widehat{r_u}$ represent the ground truth of the bins and the residual offsets.

Semi-supervised learning requires full mining of the value of unlabeled data. As a kind of semi-supervised learning, the core idea of the Consistency-based method is to reduce the consistency loss. For unlabeled data, the model should make consistent predictions.

We use a Consistency Enforcing (CE) loss [27] to ensure the consistency between the localization confidence and the classification confidence, so that boxes with high localization confidence possess high classification confidence. The consistency enforcing loss can be written as follows,

$$L_{ce} = -\log\left(C * \frac{Area(D \cap G)}{Area(D \cup G)}\right) \tag{5}$$

where $D$ and $G$ represent the predicted bounding box and the ground truth bounding box. $C$ denotes the classification confidence for the predicted bounding box. To optimize this loss function, the classification confidence and location confidence (i.e., IoU) should be maximized. Therefore, boxes with large overlaps have high classification possibilities and are kept in the Non-Maximum Suppression (NMS) procedure. The CE loss aims at ensuring the consistency between the localization and the classification to assist the NMS procedure, keeping more accurate bounding boxes.

## 4. Experimental Results

### 4.1. Datasets and Processing Platform

The KITTI dataset [34] is one of the most popular datasets and is frequently used in 3D detection for autonomous driving. There are 7481 training samples and 7518 test samples in the KITTI dataset, where the training samples are divided into the train split (3712 samples) and the validation split (3769 samples). We compare our method with state-of-the-art methods using the test split.

The experiments are performed on a standard computer under Ubuntu 16 with a core i7-6850k 3.6 GHz Central Processing Unit (CPU) with 16 GB random-access memory and a graphics processing unit (GPU), NVIDIA Titan RTX, for training and testing. Our codes are built on Pytorch which requires a Compute Unified Device Architecture (CUDA) and CUDA deep neural network library (cuDNN).

### 4.2. Implementation Details

The architecture of 3D backbone and 2D backbone follows the design in EPNet [28]. The two-stream RPN takes both the LiDAR point clouds and the camera images as input. For each 3D scene, the range of LiDAR point clouds is $[-40, 40]$, $[-1, 3]$, $[0, 70.4]$ meters along the X (right), Y (down), and Z (forward) axis in the camera coordinate, respectively. The orientation θ is in the range of $[-\pi, \pi]$. As the input for the image stream, 16,384 points are subsampled from the raw LiDAR point cloud data, which is the same as in EPNet [27]. The image stream takes images with a resolution of $1280 \times 384$ as input. We train the model for around 50 epochs with a batch size of 24 in an end-to-end manner.

### 4.3. Experiments on KITTI Dataset

We evaluated our PIFNet on KITTI Dataset following the common protocol to report the Average Precision (AP) of class Car with 0.7 (IoU) threshold. Table 1 presents quantitative results on the KITTI test set [34] for several state-of-the-art methods. Our proposed

method (PIFNet) outperforms multi-sensor-based methods MV3D [31], F-Point Net [30], MMF [24], and EPNet [27] by 20.96%, 14.31%, 6.48%, and 3.94% in terms of 3D average mAP. The results for other methods are brought from the KITTI leaderboard [34].

**Table 1.** Comparisons with state-of-the-art methods on the testing set of the KITTI dataset (Cars). L and R represent the LiDAR point cloud and the RGB image.

| Method | Modality | 3D Detection | | | | BEV Detection | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Easy | Mod. | Hard | mAP | Easy | Mod. | Hard | mAP |
| Point RCNN | L | 86.96 | 75.64 | 70.70 | 77.77 | 92.13 | 87.39 | 82.72 | 87.41 |
| Fast Point R-CNN | L | 85.29 | 77.40 | 70.24 | 77.76 | 90.87 | 87.84 | 80.52 | 86.41 |
| PV-RCNN | L | 90.25 | 81.43 | 76.82 | 82.83 | 94.98 | **90.65** | 86.14 | 90.59 |
| Voxel RCNN | L | 90.90 | 81.62 | 77.06 | 83.19 | - | - | - | - |
| MV3D | L + R | 74.97 | 63.63 | 54.00 | 64.20 | 86.62 | 78.93 | 69.80 | 78.45 |
| F-Point Net | L + R | 82.19 | 69.79 | 60.59 | 70.85 | 91.17 | 84.67 | 74.77 | 83.54 |
| MMF | L + R | 88.40 | 77.43 | 70.22 | 78.68 | 93.67 | 88.21 | **91.99** | **91.29** |
| EPNet | L + R | 89.81 | 79.28 | 74.59 | 81.22 | 94.22 | 88.47 | 83.69 | 88.79 |
| PIFNet (Ours) | L + R | **92.53** | **83.03** | **79.92** | **85.16** | **95.89** | 89.08 | 87.91 | 90.96 |

In Table 1, our method showed the best results with mAP of 85.16% in 3D detection. The second best method was EPNet [27] with mAP of 81.22%.

One can observe that the proposed PIFNet has achieved the significant performance gains over other camera-LiDAR fusion-based detectors in the KITTI leaderboard. In particular, our approach improved the average mAP by up to 4.27% compared to that of EPNet [27], the best published fusion-based method so far. While PIFNet outperforms the MMF [24] for easy and moderate cases in BEV detection, MMF [24] showed the best performance for the hard case. Compared with Lidar-only methods, PIFNet also outperforms most of 3D detectors except PV-RCNN [6] for the hard case in BEV detection. Since PV-RCNN [6] combines the point-wise feature and voxel-wise feature, it might have stronger Lidar feature.

Furthermore, we evaluated the inference time on a GPU board, and the running time is 0.21 s per frame on average. In Table 2, the running times are compared with several other methods which are based on the fusion of Lidar point clouds and camera images. The time-consuming is a bit longer than other methods because our network utilizes rather a large number of intermediate parameters and features.

**Table 2.** Running time comparisons with several camera-LiDAR fusion-based methods.

| Methods | Running Time (s/frame) |
|---|---|
| MV3D | 0.36 |
| F-Point Net | 0.17 |
| MMF | 0.08 |
| PIFNet (Ours) | 0.21 |

### *4.4. Ablation Study*

To explore the contribution of each PIFNet component, ablation experiments are performed, and the results are summarized in Table 3. We removed the EDF module and fusion module, one by one, to verify their effectiveness.

**Table 3.** Effectiveness of EDF module and fusion module.

| Method | Modality | 3D Detection | | | BEV Detection | | |
|---|---|---|---|---|---|---|---|
| | | Easy | Mod. | Hard | Easy | Mod. | Hard |
| PIFNet + Fusion | L + R | 89.22 | 81.79 | 79.15 | 91.73 | 87.41 | 87.00 |
| PIFNet + EDF | L + R | 91.45 | 82.28 | 79.46 | 93.01 | 87.96 | 87.13 |
| PIFNet + Fusion + EDF | L + R | 92.53 | 83.03 | 79.92 | 95.89 | 89.08 | 87.91 |

The fusion module can utilize the point-wise features and image-wise features by combining the two features to achieve good performance. The EDF module effectively extracted image features and boosted the performance more than that of the fusion module. It is also possible to apply our EDF module and fusion strategies to other detectors to improve their performance.

To explore the effect of the number of fusion modules on the object detection performance, we compared mAP values of architectures with different number of fusion modules. The performance results for 3 architectures with 3, 4, and 5 fusion modules are summarized in Table 4. In the experiments, the depth of EDF and the number of SA/FP layers are adjusted accordingly. As shown in Table 4, the architecture with larger number of fusion modules shows improved performance. When the number of fusion modules is increased from 3 to 4, mAP is improved by 1.73. However, when the number of fusion modules is increased from 4 to 5, mAP is improved only by 0.05, while the training time is increased significantly from 73 h to 116 h. Therefore, we choose 4 fusion modules in our normal architecture.

**Table 4.** Optimization of the number of fusion modules.

| #Fusion Modules | 3D Detection mAP | Training Time (Hour) |
|---|---|---|
| 3 | 83.43 | 51 |
| 4 | 85.16 | 73 |
| 5 | 85.21 | 116 |

### *4.5. Discussion*

In our approach, we developed an Encoder Decoder Fusion module to extract the image features and a fusion module to integrate the two stream features for accurate 3D object detection. The EDF module can fuse the low-level feature maps with high resolution and high-level feature maps with dense semantic information. The fused feature maps have accurate localization information and rich color and texture information. The fusion module is applied against the irregularity and sparsity of the point cloud, and thus, our PIFNet can outperform other state-of-the-art methods tested.

## 5. Conclusions

In this research, we have proposed a two stream Point-Image Fusion network (PIFNet) for 3D object detection to solve the problem of inconsistency between the localization and classification. We focus on fusing the depth information from point clouds generated by a Lidar and the semantic information from images generated by a camera. In order to generate the multiple scale feature maps, the Encoder-Decoder Fusion module has been designed to effectively extract image features, such as color and texture features. In addition, we align different modalities using a fusion module to generate accurate 3D proposals.

Extensive experiments demonstrated that our PIFNet can achieve good performance. Particularly, PIFNet has achieved an average mAP of 85.16% on the KITTI dataset, which is an 1.97% improvement over the previous best performance. In future works, we plan to explore more accurate methods for real-time 3D object detection solutions.

**Author Contributions:** W.Z. developed the idea and implemented the experiments. H.X. prepared the KITTI dataset and participated in ablation experiments. Y.C. participated in ablation experiments. J.R. and H.S. supervised the research and performed revisions and improvements. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under 343 Industrial Technology Innovation Program (10080619).

**Data Availability Statement:** The data in this study can be requested from the corresponding author.

**Acknowledgments:** This research was funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under 343 Industrial Technology Innovation Program (10080619).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Shuran, S.; Jianxiong, X. Sliding Shapes for 3D object detection in depth images. In Proceedings of the European Conference on Computer Vision—ECCV, Zurich, Switzerland, 5–12 September 2014.
2. Dominic, Z.W.; Ingmar, P. Voting for voting in online point cloud object detection. In Proceedings of the Robotics: Science and Systems XI, Rome, Italy, 13–17 July 2015.
3. Feng, G.; Caihong, L.; Bowen, Z. A dynamic clustering algorithm for Lidar obstacle detection of autonomous driving system. *IEEE Sens.* **2021**, *21*, 25922–25930.
4. Shuran, S.; Jianxiong, X. Deep sliding shapes for amodal 3D object detection in RGB-D. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
5. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Ipod: Intensive point-based object detector for point cloud. *arXiv* **2018**, arXiv:1812.05276.
6. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.
7. Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; Li, H. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *arXiv* **2021**, arXiv:2102.00463.
8. Jiajun, D.; Shaoshuai, S.; Peiwei, L.; Wengang, Z.; Yanyong, Z.; Houqiang, L. Voxel R-CNN: Towards high performance voxel-based 3d object detection. *arXiv* **2020**, arXiv:2012.15712.
9. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1742–1749.
10. Qi, C.R.; Hao, S.; Kaichun, M.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
11. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv* **2017**, arXiv:1706.02413.
12. Runzhou, G.; Zhuangzhuang, D.; Yihan, H.; Yu, W.; Sijia, C.; Li, H.; Yuan, L. Afdet: Anchor free one stage 3d object detection. *arXiv* **2020**, arXiv:2006.12671.
13. Chenhang, H.; Hui, Z.; Jianqiang, H.; Xian-Sheng, H.; Lei, Z. Structure aware single-stage 3d object detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11873–11882.
14. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
15. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Std: Sparse-to-dense 3d object detector for point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1951–1960.
16. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
17. Wang, Y.; Fathi, A.; Kundu, A.; Ross, D.A.; Pantofaru, C.; Funkhouser, T.; Solomon, J. Pillar-based object detection for autonomous driving. In Proceedings of the 16th European Conference of Computer Vision–ECCV, Glasgow, UK, 23–28 August 2020; pp. 18–34.
18. Shi, S.; Wang, Z.; Wang, X.; Li, H. Part-$A^2$ net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv* **2019**, arXiv:1907.03670.
19. Barrera, A.; Guindel, C.; Beltrán, J.; García, F. BirdNet+: End-to-end 3D object detection in LiDAR Bird's Eye View. In Proceedings of the IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020.
20. Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3d object detection for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
21. Bin, X.; Zhenzhong, C. Multi-level fusion based 3d object detection from monocular images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
22. Chen, X.; Kundu, K.; Zhu, Y.; Ma, H.; Fidler, S.; Urtasun, R. 3d object proposals using stereo imagery for accurate object class detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1259–1272. [CrossRef] [PubMed]

23. Peiliang, L.; Xiaozhi, C.; Shaojie, S. Stereo R-CNN based 3D Object Detection for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
24. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7345–7353.
25. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep continuous fusion for multi-sensor 3d object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
26. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3d proposal generation and object detection from view aggregation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8.
27. Huang, T.; Liu, Z.; Chen, X.; Bai, X. Epnet: Enhancing point features with image semantics for 3D object detection. In Proceedings of the European Conference on Computer Vision—ECCV, Glasgow, UK, 23–28 August 2020; pp. 35–52.
28. Pang, S.; Morris, D.; Radha, H. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 Octomber–24 January 2020; pp. 10386–10393.
29. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In Proceedings of the 16th European Conference of Computer Vision–ECCV, Glasgow, UK, 23–28 August 2020; pp. 720–736.
30. Charles, R.Q.; Wei, L.; Chenxia, W.; Hao, S.; Leonidas, J.G. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 918–927.
31. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
32. Meyer, G.P.; Charland, J.; Hegde, D.; Laddha, A.; Vallespi-Gonzalez, C. Sensor fusion for joint 3D object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
33. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
34. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.