PIFENET: PILLAR-FEATURE NETWORK FOR REAL-TIME 3D PEDESTRIAN DETECTION FROM POINT CLOUDS

Duy Tho Le, Hengcan Shi, Hamid Rezatofighi, Jianfei Cai

Department of Data Science and AI
Monash University, Australia
tho.le@monash.edu

ABSTRACT

We present PiFeNet, an efficient and accurate real-time 3D detector for pedestrian detection from point clouds. We address two challenges that 3D object detection frameworks encounter when detecting pedestrians: low expressiveness of pillar features and small occupation areas of pedestrians in point clouds. Firstly, we introduce a stackable Pillar Aware Attention (PAA) module for enhanced pillar features extraction while suppressing noises in the point clouds. By integrating multi-point-aware-pooling, point-wise, channel-wise, and task-aware attention into a simple module, the representation capabilities are boosted while requiring little additional computing resources. We also present Mini-BiFPN, a small yet effective feature network that creates bidirectional information flow and multi-level cross-scale feature fusion to better integrate multi-resolution features. Our approach is ranked 1st in KITTI pedestrian BEV and 3D leaderboards while running at 26 frames per second (FPS) and achieve state-of-the-art performance on Nuscenes detection benchmark.

Index Terms— 3D object detection, pillar attention, real-time, deep learning

1. INTRODUCTION

Light detection and ranging (LiDAR) is commonly used in machine perception tasks, especially in autonomous driving. The advancement of LiDAR technologies has encouraged the development of 3D point cloud object detection neural networks, which benefit from LiDAR’s insensitivity to weather changes and precise location of points.

3D object detection frameworks can be divided into three categories: voxel-based, point-based, and pillar-based. Point-based methods apply sampling techniques and usually use PointNet as backbone to learn point features. Voxel-based and pillar-based methods use intermediate point cloud representations such as pillars or voxels, and lightweight feature networks for feature extraction. Compared to point-based, voxel- and pillar-based methods are generally faster but less accurate.

Recent voxel- and pillar-based detectors have shown high detection efficiency for big objects such as trucks, buses, and cars. Their effectiveness in detecting pedestrian class, however, is negligible due to human distinctive characteristics: appearances in a variety of backgrounds (bushes, trees, poles, etc.), and small size (fewer points in LiDAR point cloud). These properties present two challenges:

Challenge 1: Low expressiveness in pillar features caus-
Fig. 2. Architecture of PiFeNet.

...ing pedestrians misclassified as background. Previous methods such as Pointpillars [4] struggles distinguishing pedestrians from poles and trees due to 3D spatial information loss when converting LiDAR point clouds into pillars and extracting pillar features. TANet [8] tries to solve this challenge by incorporating a triple attention module to enhance the pillar feature extraction. However, to retain important information while generalising to various object representations, the models must have more comprehensive attention mechanisms and be task-aware, knowing each channel in the pillar features accounts for different downstream tasks [9]. Furthermore, previous methods usually use max pooling to capture context information, which only shows the most forward and rightward locations in the x- and y-axes. Other points inside the pillars are ignored, but they are also critical for object localisation.

To address these problems, we propose a Pillar Aware Attention (PAA) module, including multi-point-aware pooling, point-wise, channel-wise, and task-aware attention techniques to enhance pillar features expressiveness. Multi-point-aware pooling uses various pooling strategies to grasp context information from all points in a pillar. Point- and channel-wise attentions retain proper information during the feature extraction by suppressing redundant information. Task-aware attention switches the channels on or off based on their contribution to downstream tasks such as selecting expressive channels for pseudo-image construction. Eventually, the strongest and most expressive features are chosen to form the pseudo-image. Also, PAA module can be stacked to boost the performance.

**Challenge 2: Small occupation of pedestrians in point clouds.** Prior works cannot always well detect small pedestrians in point clouds. To overcome this challenge, the feature network must have more comprehensive connections so that features from multiple resolutions can complement and compensate each other, enriching feature representations in both low- and high-level point clouds information.

Therefore, we introduce Mini-BiFPN, a lightweight feature network inspired by [10] that can bidirectionally connect and fuse multi-resolution features. Also, the learnable weights at fusion gates can be adjusted based on the contribution of features at multiple resolutions. The Mini-BiFPN is composed of three separable convolutional blocks with a minimal number of layers, making it lightweight and efficient.

Combining the above mentioned modules (as in Fig. 2), we present PiFeNet, a novel end-to-end trainable 3D object detector capable of extracting important features in real-time, which benefits from the reduced inference latency of single-stage methods and the improved feature extraction mechanisms from attention. Our contributions are as follows:

- We introduce Pillar Aware Attention Module that performs attention on pillars at the feature learning stage. Stacking PAA modules may also increase feature attention on multiple levels.
- We propose Mini-BiFPN, a lightweight feature network that efficiently performs weighted multi-level feature fusion and provides bidirectional information flow.
- We exhaustively test and compare our framework with other current state-of-the-art methods using KITTI [1] and Nuscenes [11] benchmarks. Our method is ranked 1st on both BEV and 3D pedestrian detection while running at 26 FPS on KITTI dataset (as in Fig. 1).

2. OUR APPROACH

2.1. Pillar-Feature Network (PiFeNet)

PiFeNet contains four steps. As shown in Fig. 2, the first step is quantisation, 3D point clouds are converted to pillars by a pillar generator module. In the second step, we stack our proposed PAA modules that extract features of pillars with reduced information loss. The third step involves scattering the learned features onto a pseudo-image and feeding it into our Mini-BiFPN to fuse features at different resolutions. Finally, we use a simple detection head including classification and box regression branches to produce the detection results.

2.2. Pillar Aware Attention (PAA) module

As shown in Fig. 3, PAA module begins with multi-point-aware pooling followed by two branches of point- and
pressive and sensitive to all of the points in the point clouds attention sub-modules, the pillar features become more ex-
we can obtain attention-weighted features in both channel-
M using element-wise multiplication
R
F
A
element-wise multiplication before going through the task-
channel-wise attention. Then the outputs are integrated using
tinct channel context representations
to aggregate channel information, which results in two dis-
N
Fig. 3. Architecture of Pillar Aware Attention module.
channel-wise attention. Then the outputs are integrated using
element-wise multiplication before going through the task-
aware attention sub-module.
Multi-point-aware pooling. To capture the context of all
points and channels in a pillar, we present multi-point-
aware pooling, which applies both max and average pooling to the point- and channel-wise dimensions in point clouds. For a pillar $P_j \in \mathbb{R}^{P \times N \times C}$ in the pillar grid, where $N$ the maximum number of points and $C$ is the number of channels in a pillar, and $P$ is the amount of pillars in the grid where $P = \{p_1, p_2, \ldots, p_N\} \subseteq \mathbb{R}^{12000}$. We perform average pooling and max pooling across $N$ points in the pillar to aggregate channel information, which results in two distinct channel context representations $F_{c, mean}^c$ and $F_{c, max}^c$ where $F_{c, mean}^c, F_{c, max}^c \in \mathbb{R}^{P \times 1 \times C}$. The same strategies are applied to the point-wise dimension to aggregate the point information $F_{p, mean}^c$ and $F_{p, max}^c$ where $F_{p, max}^c, F_{p, mean}^c \in \mathbb{R}^{P \times N \times 1}$

Channel- and Point-wise attention. To encapsulate the global information, $F_{c, mean}^c$ and $F_{c, max}^c$ are passed through a shared multi-layer perceptron (MLP) network with two fully connected layers, an activation function, and a reduction ratio $r$ (the same applied to $F_{p, mean}^c$ and $F_{p, max}^c$). The shared MLP’s outputs are summed element-wise to generate the final attention score vector $A_c \in \mathbb{R}^{P \times 1 \times C}$ and $A_p \in \mathbb{R}^{P \times N \times 1}$:

$$A_{c/p} = \sigma \left( w_1 \left( w_0 \left( F_{c/p, mean}^c \right) \right) + w_1 \left( w_0 \left( F_{c/p, max}^c \right) \right) \right)$$

where $\sigma$ is the sigmoid function, $w_0 \in \mathbb{R}^{N/r \times C}$ and $w_1 \in \mathbb{R}^{C \times N/r}$ are the weights in the two fully-connected layers. The full attention matrix is then achieved by combining channel-wise attention $A_c$ and point-wise attention $A_p$ using element-wise multiplication $M_j = A_p \times A_c$, where $M_j \in \mathbb{R}^{N \times P \times C}$. By multiplying $M_j$ by the original pillar $P_j$, we can obtain attention-weighted features in both channel-wise and point-wise dimensions.

After being processed by the point-wise and channel-wise attention sub-modules, the pillar features become more expressive and sensitive to all of the points in the point clouds and their channel features ($x$, $y$, $z$ locations, centres).

Task-aware attention. The task-aware attention sub-module is chained at the end to dynamically control the activation of each channel, the output pillar features are now reorganised into distinct activations in response to the needs of various downstream tasks. Similar to [12], given a feature map $F_c$ of a channel $C_k$ in the pillar $P_j$ and $[\alpha_1, \beta_1, \alpha_2, \beta_2]^T$ are the learnable parameters to monitor the activation functions. The activation at the channel $C_k$ is calculated as follow:

$$A_k^T(F_c) = \max_{i \in \{1, 2\}} \left\{ \alpha_i^k \cdot F_c + \beta_i^k \right\}$$

where $\alpha_i^k$ and $\beta_i^k$ are the $\alpha$ and $\beta$ at the $k$th channel and $A_k^T(F_c)$ is the task-aware weighted pillar feature map of the channel $C_k$

To regulate the activation thresholds, a hyper function $\theta(\cdot) = [\alpha_1, \beta_1, \alpha_2, \beta_2]$ is used where $[\alpha_1, \beta_1, \alpha_2, \beta_2]$ is initialized with [1, 0, 0, 0]. $\theta(\cdot)$ is calculated with using two fully connected layers similar to [13]. The output is normalized by a shifted sigmoid layer $f(x) = 2\sigma(x) - 1$. Finally, the task-aware weighted features are concatenated or summed back to the original features.

Stackable PAA module. Our approach stacks two PAA modules to better leverage multi-level feature attention. The first module concatenates nine task-aware weighted features with the original ones; then the resulting 18 channels are increased to 64 at the end of the second PAA module. Eventually, a point-wise max pooling is performed to extract the strongest features, serving as inputs of Mini-BiFPN module, as shown in Fig. 2

2.3. Mini-BiFPN

We develop Mini-BiFPN detection head, a mini variant of BiFPN [10] that can greatly improve the model’s performance in 3D object detection with minimal efficiency trade-offs. The Mini-BiFPN firstly takes a pseudo-image, passes it through convolutional blocks $B_1, B_2, B_3$ to produce a list of features $\tilde{F}_{in} = (F_{in}^1, F_{in}^2, F_{in}^3)$, with resolutions $1/2^{i-1}$ of input pseudo-image where $i \in \{1, 2, 3\}$. Next the multi-scale features are aggregated by repeatedly applying top-down and bottom-up bidirectional feature fusion, as shown in Fig. 4
Trainable weights are added to adjust the fusion weights accordingly. Our Mini-BiFPN can be formulized as follows:

$$F^2_{up} = \text{conv} \left( \text{swish} \left( \frac{w_i F^2_{in} + w'_i \text{upsample} (F^3_{in})}{w_i + w'_i + \epsilon} \right) \right)$$  \hspace{1cm} (3)

$$F^2_{out} = \text{conv} \left( \text{swish} \left( \frac{w_i F^2_{in} + w'_i F^3_{in} + w''_i \text{downsample} (F^3_{in})}{w_i + w'_i + w''_i + \epsilon} \right) \right)$$  \hspace{1cm} (4)

where $F^2_{up}$ is the fusion result of $F^2_{in}$, $F^3_{in}$. Then $F^2_{out}$ is calculated using $F^2_{up}$, $F^1_{out}$, and $F^3_{out}$. $w_i$, $w'_i$, and $w''_i$ are trainable parameters, where $i$ is the number of fused features at a particular block. $\text{swish}$ is Swish [21] activation function. $F^1_{out}$ and $F^3_{out}$ are computed similarly to $F^2_{out}$. The final feature representation is the concatenation of $F^1_{out}$, $F^2_{out}$, and $F^3_{out}$. It is then passed into an SSD [22] detection head including classification and box regression branches to predict the detection results.

### 3. EXPERIMENTS

Our proposed PiFeNet is evaluated using KITTI [1] and Nuscenes [11] object detection benchmarks. For KITTI dataset, we choose PointPillars [4] as our baseline, and we adopt the training and testing configurations from SECOND [7]. For Nuscenes dataset, we choose CBGS [23] as our baseline. We also reproduce their results (CBGS-PP with Pointpillars backbone) for an apple to apple comparison.

#### 3.1. Implementation Details

For KITTI dataset, we split the data with a ratio of 0.85:0.15 and train the network on an NVIDIA Titan V100 with batch size 2 for 150 epochs, using AdamW [24] optimizer with max LR of $2.25e^{-3}$, One Cycle policy [25].

For Nuscenes dataset, we adopt CBGS [23] augmentations strategies and multi-task heads, then train the model for 20 epochs with batch size of 2, and $2e^{-3}$ max LR. During testing, 83 predictions with confidence scores greater than 0.1 are kept, and rotated NMS has IOU threshold of 0.02.
3.2. Results and Comparison with State-of-the-arts

KITTI detection benchmark. As in Tab. 1, PiFeNet surpasses all current state-of-the-art approaches in both BEV and 3D detection. Our model outperforms TANet [8] by 3.10 mAP in BEV and 1.4 mAP in 3D detection. PiFeNet exceeds the best BEV pedestrian detector Frustum-PointPillars [14] by 2.53 mAP. Also, PiFeNet exceeds the current top 3D pedestrian detector HotSpotNet [17] by 0.94 mAP while running faster.

NuScenes detection benchmark. According to Tab. 2, without any ensembling or configuration setting tricks, our real-time method outperforms other state-of-the-art approaches by 1.2% in pedestrian detection, and achieves the highest overall mAP and NDS. On the validation set, PiFeNet reaches 0.48 mAP/ 0.61 NDS compared to CBGS-PP (0.42 mAP/ 0.56 NDS).

Qualitative Analysis. Our qualitative findings are shown in Fig. 5 (A and C), with common failure scenarios (B and D). Fig. 5C demonstrates our model’s performance by accurately detecting all pedestrians in the image, even at great distances. Although the model rarely misidentifies a tree/pole as a pedestrian, we did discover an occasion where the model was confused by a stack of soda crates (Fig. 5D). As shown in (Fig. 5B), the model cannot reliably locate both pedestrians when they are too close together, and in most situations, one of the two people is ignored.

3.3. Ablation Studies

Attention modules analysis. According to Tab. 3, channel-wise and task-aware attention alone does not significantly improve model performance. However, combining them results in a significant 2.2% mAP increase in pedestrian detection. Also, combining point-wise attention with neither channel-wise nor task-aware would improve the performance. Therefore, while standalone point-wise attention (2.3% mAP increase) can significantly increase model performance, channel-wise and task-aware sub-modules must be employed together for better accuracy.

Pooling mechanisms analysis. In Tab. 3, we show that each pooling method improves the model’s performance. The z-axis related channels may be used to extract the maximum height of points inside a pillar, while the x and y-axis related channels can be used to extract the mean position of points inside a pillar. While average pooling outperforms max pooling in detecting moderate and hard objects, max pooling surpasses average pooling in detecting moderate and hard items. Concurrent usage of both pooling methods increases mAP to 64.2, showing that both pooling mechanisms are required.

Mini-BiFPN module analysis. As in Tab. 3, switching from the baseline feature network to the Mini-BiFPN module improves mAP by 6.23 percent, proving that Mini-BiFPN is a key part of our final model.
4. CONCLUSION

Our PiFeNet has been presented, addressing the two main issues in 3D pedestrian detection. First, we introduced the Pillar Aware Attention Module, which combines multi-point-aware-pooling, point-wise, channel-wise, and task-aware attention to better extract pillar features. Next is the Mini-BiFPN module, a lightweight feature network that leverages cross-scale feature fusion and bidirectional connections, enriching information flow in the feature network. Our method achieves state-of-the-art performance on both large scale benchmarks KITTI [1] and Nuscenes [11].

5. REFERENCES

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, “Vision meets robotics: The kitti dataset,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.
[2] Xinchen Ye, Mingliang Zhang, Rui Xu, Wei Zhong, Xin Fan, Zhu Liu, and Jiaao Zhang, “Unsupervised monocular depth estimation based on dual attention mechanism and depth-aware loss,” in ICME. IEEE, 2019, pp. 169–174.
[3] Yufei Wei, Yao Xiao, Yibo Guo, Shichao Liu, and Lin Xu, “Lpcc-net: Rgb guided local point cloud completion for outdoor 3d object detection,” in ICME. IEEE, 2021, pp. 1–6.
[4] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in CVPR, 2019, pp. 12697–12705.
[5] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu, “Se-ssd: Self-ensembling single-stage object detector from point cloud,” in CVPR, 2021, pp. 14494–14503.
[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in CVPR, 2017, pp. 652–660.
[7] Yan Yan, Yuxing Mao, and Bo Li, “Second: Sparsely embedded convolutional detection,” Sensors, vol. 18, no. 10, pp. 3337, 2018.
[8] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai, “Tanet: Robust 3d object detection from point clouds with triple attention,” in AAAI, 2020, vol. 34, pp. 11677–11684.
[9] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang, “Dynamic head: Unifying object detection heads with attentions,” in CVPR, 2021, pp. 7373–7382.
[10] Mingxing Tan, Ruoming Pang, and Quoc V Le, “Efficientdet: Scalable and efficient object detection,” in CVPR, 2020, pp. 10781–10790.
[11] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in CVPR, 2020, pp. 11621–11631.
[12] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu, “Dynamic relu,” in ECCV. Springer, 2020, pp. 351–367.
[13] Jie Hu, Li Shen, and Gang Sun, “Squeeze-and-excitation networks,” in CVPR, 2018, pp. 7132–7141.
[14] Anshul Paigwar, David Sierra-Gonzalez, Ozgur Erkent, and Christian Laugier, “Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar,” in CVPR, 2021, pp. 2926–2933.
[15] Jiale Li, Hang Dai, Ling Shao, and Yong Ding, “From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder,” arXiv:2108.03648, 2021.
[16] Shaoshuai Shi, Chaoux Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaoang Wang, and Hongsheng Li, “Pv-rcnn: Point-voxel feature set abstraction for 3d object detection,” in CVPR, 2020, pp. 10529–10538.
[17] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille, “Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots,” in ECCV. Springer, 2020, pp. 68–84.
[18] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan, “What you see is what you get: Exploiting visibility for 3d object detection,” in CVPR, 2020, pp. 11001–11009.
[19] Yangyang Ye, Houjin Chen, Chi Zhang, Xiaoli Hao, and Zhaoxiang Zhang, “Sarpnet: Shape attention regional proposal network for lidar-based 3d object detection,” Neurocomputing, vol. 379, pp. 53–63, 2020.
[20] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki, “Sa-det3d: Self-attention based context-aware 3d object detection,” arXiv:2101.02672, 2021.
[21] Prajit Ramachandran, Barret Zoph, and Quoc V Le, “Searching for activation functions,” arXiv:1710.05941, 2017.
[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, “Ssd: Single shot multibox detector,” in ECCV. Springer, 2016, pp. 21–37.
[23] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu, “Class-balanced grouping and sampling for point cloud 3d object detection,” arXiv:1908.09492, 2019.
[24] Ilya Loshchilov and Frank Hutter, “Decoupled weight decay regularization,” arXiv:1908.03704, 2019.
[25] Leslie N Smith and Nicholas Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in Artificial Intelligence and
Machine Learning for Multi-Domain Operations Applications. International Society for Optics and Photonics, 2019, vol. 11006, p. 1100612.