



MSCSeg: Multi-scale contextual network for LiDAR point cloud semantic segmentation

Yan Zhou¹ · Yichao Fan¹ · Haibin Zhou² · Richard Irampaye²

Received: 4 January 2025 / Revised: 16 June 2025 / Accepted: 27 June 2025 / Published online: 10 September 2025
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2025

Abstract

The LiDAR point cloud semantic segmentation task aims to achieve 3D environmental perception of the surrounding scene. Range image-based methods have shown promising prospects for real-time processing in the field of autonomous driving. However, the projection process of generating range images has the problem of stretching the spatial distance relationship between the original points. Furthermore, most existing works lack the ability to effectively capture the target's boundary features and cannot compensate for the loss of high-resolution encoder features during downsampling. To address these issues, we propose MSCSeg, a range image-based multi-scale contextual network for point cloud semantic segmentation. Firstly, a branched asymmetric convolution residual block (BAResBlock) is employed to specifically process the stretched range image while aggregating multi-scale contextual information from three branches at low computational cost. Subsequently, we introduce a local context multi-scale strip convolution block (CSCB), which helps the network capture boundary features of objects. Finally, we employ a feature cascade attention block (FCAB) to aggregate information from the encoder and decoder, alleviating the problem of insufficient detail in low-resolution features. Experiments conducted on the outdoor street scene datasets SemanticPOSS and SemanticKITTI demonstrate that our MSCSeg achieves competitive segmentation performance while maintaining real-time inference speed.

Keywords LiDAR point cloud · 3D semantic segmentation · Range image-based · Autonomous driving

1 Introduction

Autonomous driving is one of the most promising application scenarios for computer vision technology. Among the various in-vehicle sensors used in practical applications, LiDAR is favoured for navigation tasks due to its ability to accurately capture 3D spatial information of the surrounding environment and generate point cloud representations [1, 2]. Accurate and real-time semantic segmentation of the point cloud is a core task in the perception system of autonomous vehicles [3–5], and plays a crucial role in path planning [6–8].

With the emergence of large-scale point cloud datasets and the rapid advancement of computing power, deep learning has gradually come to dominate the field of point

cloud semantic segmentation [9, 10]. Deep learning-based point cloud semantic segmentation methods can be divided into three types: point-based methods [11–15], voxel-based methods [16–18] and projection-based methods [19–31]. Projection-based methods primarily fall into two main categories: multi-view and spherical projection-based methods. This paper focuses on the latter. Current spherical projection-based methods suffer from the following issues:

- During the spherical projection of point clouds, the spatial distance relationships between the original points are non-linearly stretched in the generated range image due to the inherent characteristics of the spherical projection, as shown in Fig. 1. This problem is more pronounced for the 360° horizontal field of view of LiDAR. Therefore, the horizontal receptive field in the range image deserves greater attention.
- Boundary blur is not uncommon in segmented objects, where boundary semantics are easily assimilated by the semantic information of surrounding points. However, in autonomous driving scenarios, it is crucial to accurately

✉ Yan Zhou
yanzhou@xtu.edu.cn

¹ School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China

² School of Mathematics and Computer Science, Xiangtan University, Xiangtan 411105, China

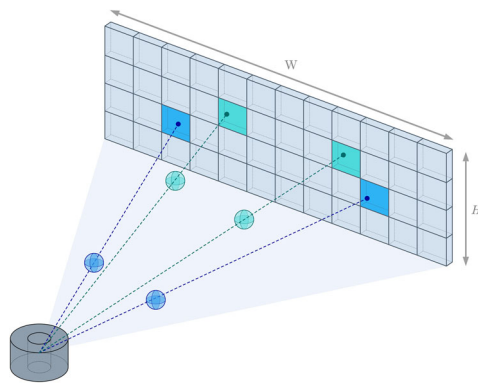


Fig. 1 The spherical projection method stretches the spatial distance relationship between the original points

identify object boundaries and assign correct semantic labels.

- Semantic segmentation is a dense prediction task that necessitates the delineation of minute details. The commonly used bilinear interpolation in the upsampling process can only generate semantically consistent features but cannot compensate for the high-resolution encoder details lost during downsampling.

In order to address the issues mentioned above, this paper proposes a range image-based point cloud semantic segmentation network. Specifically, we first designed the branched asymmetric convolution residual block (BAREs-Block). Compared with the conventional ResBlock, three branches of asymmetric depthwise separable convolutions were incorporated to replace the traditional square convolution while capturing features at different levels, thus alleviating the problem of the impact of the stretched spatial relationship of the point cloud on feature extraction. Next, we designed the local context multi-scale strip convolution block (CSCB) as the decoder. A local context window self-attention (WinSA) was introduced to process the input features, enabling the capture of local contextual information without significantly increasing the computational cost. Concurrently, to effectively mitigate the issue of boundary blurring, a multi-scale strip convolution (StrConv) was incorporated. The strip convolution kernel is extended in a single direction, enabling the model to capture information about changes in that direction, thereby facilitating the identification of the object boundary and generating more accurate predictions. In light of the insufficient detail in the encoder's deep features resulting from the low-resolution and the bilinear interpolation method's inability to restore the loss of detail caused by downsampling effectively, we developed the feature cascade attention block (FCAB). This block employs a cascaded operation, which incorporates channel and spatial attention mechanisms, to integrate information from the

encoder and decoder, thereby addressing the need for accurate detail in dense prediction.

In summary, our main contributions can be summarised as follows:

- We designed the BAREsBlock to specifically mitigate the issue of stretching the spatial relationship of the range image and to capture contextual information at varying levels with lower computational cost.
- We designed the CSCB, which consists of a window self-attention module capable of capturing fine-grained features within a local window and a strip convolution module sensitive to significant changes in coordinates, intensity, and distance. This allows the block to capture local feature details and boundary information associated with objects.
- We designed the FCAB to fuse information from the encoder and decoder to alleviate the issue of detail loss in low-resolution features.

2 Related Work

2.1 Range image-based network

SqueezeSeg [19] pioneers a range image-based method for point clouds semantic segmentation. Its encoder uses the lightweight SqueezeNet [32] architecture for efficient inference and introduces a Conditional Random Field (CRF) post-processing technique to refine 3D predictions and mitigate the "many-to-one" problem, where identical labels are assigned to multiple points mapped to the same pixel. However, SqueezeSeg suffers from limited receptive fields in complex scenes and struggles to capture long-range contextual dependencies. To address this, SqueezeSegV2 [20] improves the model architecture, training loss and normalisation methods, while proposing a Context Aggregation Module (CAM) that fuses multi-scale features through parallel dilated convolutions, significantly improving the perception of multi-scale targets. SqueezeSegV3 [23] advances Spatially Adaptive Convolution (SAC) by dynamically adjusting convolution kernel weights based on local geometric structures and texture variations in input images, improving generalisation in complex scenes. RangeNet++ [21] uses a modified DarkNet [33] as its backbone and employs an efficient K-nearest neighbor (KNN) algorithm for post-processing computations to deal with discretization errors. KNN is still a widely used post-processing method for image-based point cloud segmentation. SalsaNext [24] progressively expands receptive fields via dilated convolutions to improve multi-scale context awareness in complex scenes while integrating Bayesian processing to quantify both epistemic and aleatory uncertainties for each point. FIDNet [26]

constructs its feature extraction network using ResNet34 [34], replaces transposed convolution for upsampling with bilinear interpolation to mitigate artefacts, and proposes Nearest Label Assignment (NLA) to reduce misclassification caused by many-to-one mapping. CENet [28] builds on FIDNet by replacing MLP with standard 3×3 convolutions, introducing an auxiliary loss to address pixel misalignment resulting from bilinear interpolation, and incorporating additional nonlinear activations. These modifications increase accuracy without increasing parameters.

2.2 Lightweight Feature Extractor

In recent years, many lightweight models have introduced special convolution paradigms, such as group convolution, pointwise convolution, dilated convolution, separable convolution, and so on, to capture a wider range of feature information without significantly increasing the number of parameters and computations.

SqueezeNet [32] proposes a sparse convolution module that includes two operations: squeeze and expand. The squeeze operation uses 1×1 convolutions to reduce the number of channels in the feature map, thereby reducing the amount of computation and the number of parameters. The expand operation combines 1×1 and 3×3 convolutions to expand the number of channels in the feature map and allow the model to learn richer feature representations. MobileNets [35] uses a combination of depthwise separable convolutions and 1×1 pointwise convolutions to effectively reduce the computational complexity of the model to match the limited processing power of mobile devices [36]. MobileNetV2 [37] uses ResNet's bottleneck module, which combines depthwise separable convolutions with residual connections. ESPNet [38] applies 1×1 convolutions to reduce the number of input channels and then uses multiple dilated convolutions of different scales to capture multi-scale contextual information, thereby expanding the receptive field without increasing the computational load, memory and power consumption, which greatly facilitates semantic segmentation of high-resolution images. Compared to ESPNet, ESPNetV2 [39] uses pointwise group convolutions and depthwise separable convolutions and performs more efficiently on various tasks. CTAFNet [40] designs a Convolutional Transformation Adaptive Fusion Kernel (CTAFFK) that leverages dual CNN and Transformer branches to extract local and global features from images, achieving comprehensive feature representation while maintaining high computational efficiency.

3 Method

Spherical projection is a classic projection method, the LiDAR point cloud is projected onto a sphere to generate

a range image, as illustrated by the following Eq. 1:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2} [1 - \arctan(\frac{y}{x}) \pi^{-1}] W \\ [1 - (\arcsin(\frac{z}{r}) + fov_{up}) fov^{-1}] H \end{pmatrix} \quad (1)$$

Where (u, v) represent the coordinates of each projected point on the range image, (x, y, z) are the Cartesian coordinates of the corresponding point, and (H, W) are the desired height and width of the range image. The vertical field of view (fov) of the LiDAR is divided into an upper and a lower part, which are designated as fov_{up} and fov_{down} , respectively. The variable r represents the distance from each point to the origin. This equation establishes a correspondence between the three-dimensional point coordinates and the pixel coordinates of the range image. In the field of 2D image processing, Dp-M3D [41] effectively leverages monocular depth estimation to generate depth maps, which have proven valuable for providing depth-aware information in subsequent feature fusion and contributing to competitive performance. Our range image, in contrast, preserves the complex characteristics inherent in the original point cloud data. The range image typically has five channels, each representing a specific piece of characteristic information about the point's coordinates, range, and intensity.

The overall architecture of our MSCSeg is illustrated in Fig. 2. We adopted the classical encoder-decoder architecture. Initially, the 5-channel range image features generated by the spherical projection are fed into three standard 3×3 convolutional layers to generate a 128-channel high-dimensional tensor. Subsequently, the data is then fed into an encoder comprising four BAREsBlocks in series to generate a multi-resolution feature image, and the final three residual blocks perform downsampling of the feature map while simultaneously extracting the features. After reducing the size of the feature map to $(H/8, W/8)$, it is passed to the decoder stage of the network. The features are further processed by the CSCB. The system comprises two modules, the WinSA and the StrConv, which are connected in series. The WinSA is responsible for capturing local contextual information within the image, while the StrConv reacts significantly to drastic changes in the coordinates, distance and intensity, and is used to capture the boundary information of the target. Moreover, due to the copious detail information present in the encoder features, we integrate the multi-stage encoder features with the decoder features generated by the CSCB via the FCAB to address the issue of insufficient detail information in the low-resolution decoder features resulting from downsampling. In the end, we designed SegHead to retain the feature information of each stage and output semantic labels and used KNN for post-processing. The following section provides a comprehensive description of the modules.

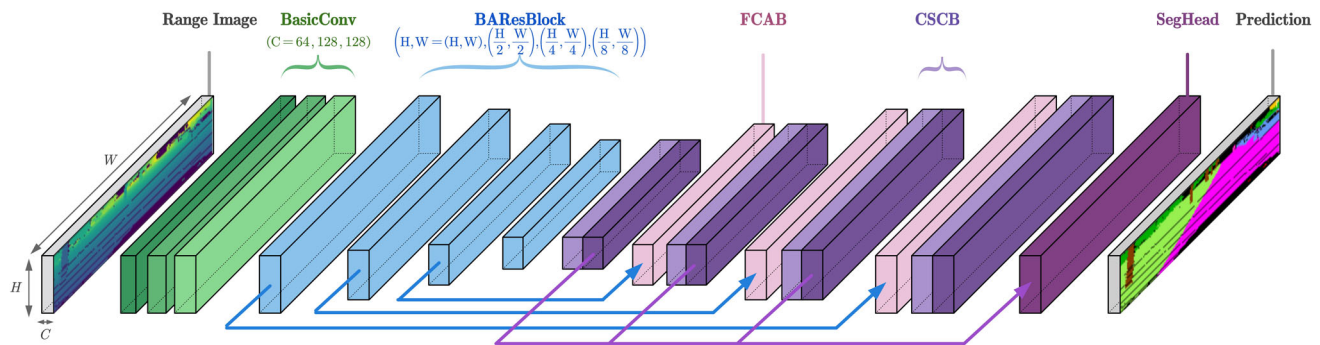


Fig. 2 The architecture of our MSCSeg

3.1 BAResBlock

The structure of our BAResBlock is shown in Fig. 3. The range image generated by spherical projection has the problem of stretching the spatial relationship between the original points. The elevation angle of the vehicle radar is limited, for example, the 64-line vehicle LIDAR has only 64 discontinuous laser beams in the vertical direction while the azimuth angle is continuous at 360° , so more attention to the horizontally oriented sensor field of the range image is required to alleviate this problem. We optimize the standard ResBlock to make it more suitable for range image segmentation tasks. We replace the original standard 3×3 convolution layer with asymmetric depthwise separable convolution modules with different scales (3×5 , 5×7 , 7×9) in three branches. This modification enables the residual module to integrate contextual information of varying scales and to possess a more extensive receptive field in the azimuth direction. It is notable that the lightweight performance is attained even with the incorporation of three convolutional layers since the depthwise separable convolutions have a small number of parameters. To minimize the impact on the speed of model inference, we only replace the first ResBlock of each encoding layer.

3.2 CSCB

We introduce the CSCB to process the feature images further. As shown in Fig. 4, CSCB consists of the WinSA for capturing local features and the StrConv for enhancing boundary features. The CSCB($\cdot$) can be formulated as in Eq. 2:

$$\text{CSCB}(x) = \text{StrConv}(\text{BN}(\text{WinSA}(\text{BN}(x)))) \quad (2)$$

3.2.1 WinSA

The WinSA captures local information in context by dividing the feature image into local windows and locally calculat-

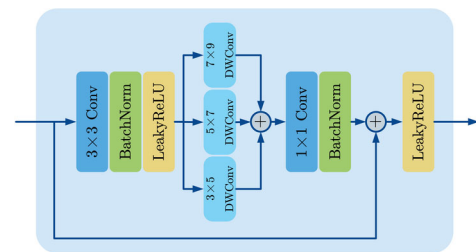


Fig. 3 The branched asymmetric convolution residual block (BAResBlock)

ing the self-attention score within the window. For a given input feature map, the size is first adjusted to align with the specified window size, and then the map is divided into multiple local windows. In each local window, Query (Q), Key (K), and Value (V) Matrices are generated. The self-attention mechanism is employed to ascertain the relevance of each token in relation to its neighboring tokens, thereby deriving an attention score. Subsequently, the score is scaled, and a relative position bias is added, after which softmax $\sigma(\cdot)$ is performed to obtain the attention weight. The weight is applied to the V matrix to generate a weighted feature. In order to control the computational cost, this process is performed independently in each window, focusing only on local information within the window. Once the calculation is complete, the results are rearranged and fused with the original input. Finally, a 3×3 depthwise convolutional layer $\text{DWC}(\cdot)$ is used to obtain the final output. The $\text{WinSA}(\cdot)$ can be formulated as in Eq. 3:

$$\text{WinSA}(x) = \text{DWC}(x + \sigma(Q \times K) \times V) \quad (3)$$

3.2.2 StrConv

The StrConv captures multi-scale features by concatenating strip convolutions of different sizes and improves the segmentation results of object boundaries. Specifically, we first use a standard 1×1 convolution $\text{C1}(\cdot)$ to preprocess the input, and then extract multi-scale features by using three branches

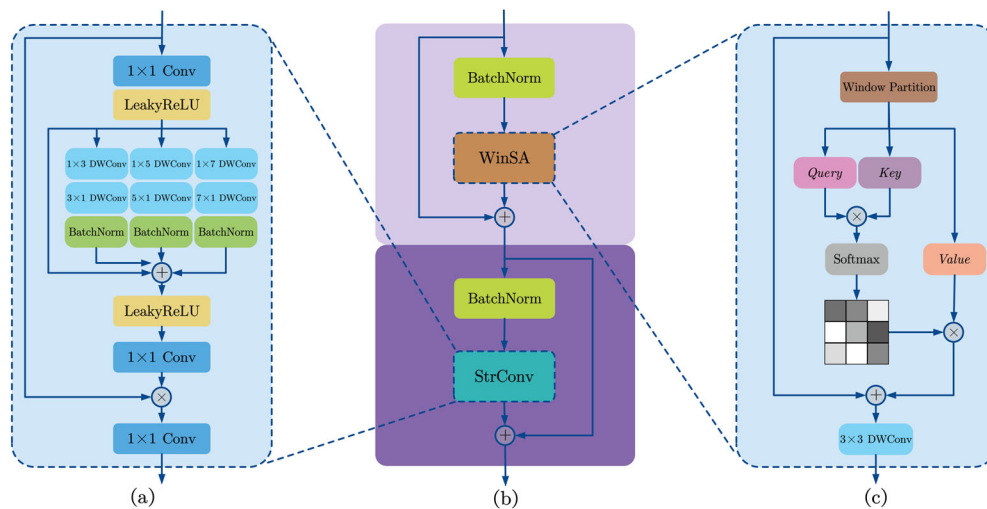


Fig. 4 (a) Multi-scale strip convolution (StrConv), (b) Local context multi-scale strip convolution block (CSCB), (c) Local context window self-attention (WinSA)

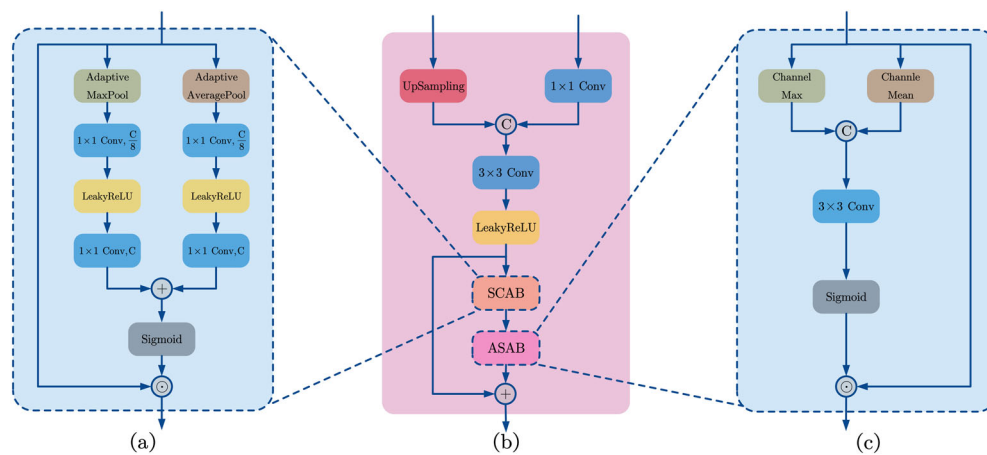


Fig. 5 (a) Squeeze channel attention block (SCAB), (b) Feature cascade attention block (FCAB), (c) Adaptive spatial attention block (ASAB)

of asymmetric strip convolution. These multi-scale feature maps are fused with the preprocessed features through a residual connection, and the original input features are further integrated with the fused features through a standard 1×1 convolution layer $C2(\cdot)$. The StrConv then enhances the model's focus on important features by element-wise multiplying this output with the original input features. The lightweight strip convolution is dilated in one dimension and compact in the other, making it sensitive to feature changes in specific directions in the image. The StrConv($\cdot$) can be formulated as in Eq. 4c:

$$F_1(x) = \text{LR}(C1(x)) \quad (4a)$$

$$F_2(x) = \sum \text{BN}(C_{i \times 1}(C_{1 \times i}(F_1(x)))) \quad (4b)$$

$$\text{StrConv}(x) = C3(C2(\text{LR}(F_1(x) + F_2(x))))x \quad (4c)$$

3.3 FCAB

The FCAB is used to fuse the channel and spatial dimensions of high-resolution encoder features and low-resolution decoder features. Its structure is shown in Fig. 5. First, the decoder feature x_{de} is upsampled by $\text{Up}(\cdot)$, and the encoder feature x_{en} is input into the 1×1 convolution channel $C1(\cdot)$. The two are then fused by $\text{CAT}(\cdot)$ to form a concatenated feature. A 3×3 convolution $C2(\cdot)$ further integrates this concatenated feature and is then fed into the squeeze channel attention block $\text{SCAB}(\cdot)$, which assigns different weights to each channel, distributing the weights to emphasize feature channels that are more relevant to the segmentation task, while suppressing useless and redundant feature channels. The output of SCAB is then fed into the adaptive spatial attention block $\text{ASAB}(\cdot)$ to identify areas in the feature map that require focused attention and to further capture local contextual information to improve the model's spatial per-

ception. Finally, the output features are residual fused with the concatenated features. The FCAB($\cdot$) can be formulated as in Eq. 5b:

$$F_3(x_{en}, x_{de}) = \text{CAT}(\text{C1}(x_{en}), \text{Up}(x_{de})) \quad (5a)$$

$$\text{FCAB}(x_{en}, x_{de}) = \text{LR}(\text{C2}(F_3(x_{en}, x_{de}))) + \text{ASAB}(\text{SCAB}(\text{LR}(\text{C2}(F_3(x_{en}, x_{de})))))) \quad (5b)$$

3.3.1 SCAB

In SCAB, the initial step involves the application of adaptive max pooling AMP($\cdot$) and adaptive average pooling AAP($\cdot$) to the input features in the spatial dimension. Adaptive max pooling is used to preserve the salient features of the image, while adaptive average pooling is used to smooth the overall distribution of the input feature map. Next, the two branches are processed concurrently through a 1×1 convolution C1($\cdot$), which reduces the number of channels to one-eighth of the original. Following applying the LeakyReLU activation function LR($\cdot$), the number of channels is restored to the original value using a 1×1 convolution C2($\cdot$). This process effectively reduces the computational complexity of the convolution. Subsequently, the features of the two branches are combined, and the attention weights are calculated using the sigmoid function $\sigma(\cdot)$. We incorporate these weights to input using the Hadamard product ($\odot$). The SCAB($\cdot$) can be formulated as in Eq. 6c:

$$F_4(x) = \text{C2}(\text{LR}(\text{C1}(\text{AMP}(x)))) \quad (6a)$$

$$F_5(x) = \text{C2}(\text{LR}(\text{C1}(\text{AAP}(x)))) \quad (6b)$$

$$\text{SCAB}(x) = \sigma(F_4(x) + F_5(x)) \odot x \quad (6c)$$

3.3.2 ASAB

In ASAB, the first step is to apply global max pooling GMP($\cdot$) and global average pooling GAP($\cdot$) operations on the input features along the channel dimension to compute the maximum and average values of the feature map in the channel dimension, which are then merged to form a 2-channel feature map. Subsequently, a 3×3 convolution C($\cdot$) is employed to capture local contextual information and enhance the local relationship between features. Analogous to SCAB, after calculating the attention weight using the sigmoid function $\sigma(\cdot)$, we perform a Hadamard product ($\odot$) between the weight and the input to obtain the output. The ASAB($\cdot$) can be formulated as in Eq. 7:

$$\text{ASAB}(x) = (\text{C}(\text{CAT}(\text{GMP}(x), \text{GAP}(x)))) \odot x \quad (7)$$

Table 1 Experimental results of our proposed MSCSeg and other LiDAR segmentation methods on the SemanticPOSS test set

method	mIoU	person	rider	car	truck	plants	traffic sign	pole	trashcan	building	cone/stone	fence	bike	ground
SqueezeSeg [19]	18.9	14.2	1.0	13.2	10.4	28.0	5.1	5.7	2.3	43.6	0.2	15.6	31.0	75.0
SqueezeSegV2 [20]	30.0	48.0	9.4	48.5	11.3	50.1	6.7	6.2	14.8	60.4	5.2	22.1	36.1	71.3
RangeNet++ [21]	30.9	57.3	4.6	35.0	14.1	58.3	3.9	6.9	24.1	66.1	6.6	23.4	28.6	73.5
SqueezeSegV3 [23]	40.0	62.2	23.1	66.1	21.2	63.0	20.5	17.8	13.3	70.9	21.7	17.5	50.3	72.7
MINet [29]	43.2	62.4	12.1	63.8	22.3	68.6	16.7	30.1	28.9	75.1	28.6	32.2	44.9	76.3
FIDNet [26]	46.4	72.2	23.1	72.7	23.0	68.0	22.2	28.6	16.3	73.1	34.0	40.9	50.3	79.1
CENet [28]	50.3	75.5	22.0	77.6	25.3	72.2	18.2	31.5	48.1	76.3	27.7	47.7	51.4	80.3
TENet [31]	51.9	72.4	20.5	77.7	24.8	71.6	29.1	37.8	46.3	79.9	34.5	47.3	53.9	78.4
LENet [30]	53.4	76.2	21.5	80.6	30.3	73.7	26.8	33.7	55.8	80.1	34.4	49.0	52.4	79.9
MSCSeg (Ours)	54.4	77.7	29.2	81.4	25.3	73.4	25.9	37.7	47.8	79.0	47.4	49.3	53.8	79.3

Table 2 Experimental results of our proposed MSCSeg and other LiDAR segmentation methods on the SemanticKITTI test set

method	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
SqueezeSeg [19]	30.8	68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	4.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3
SqueezeSegV2 [20]	39.7	81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3
RandLA-Net [13]	50.3	94.0	19.8	21.4	42.7	38.7	47.5	48.8	4.6	90.4	56.9	67.9	15.5	81.1	49.7	78.3	60.3	59.0	44.2	38.1
RangeNet++ [21]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
PolarNet [22]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
PCSCNet [18]	54.3	94.3	23.1	26.2	26.1	36.7	43.1	36.4	7.9	91.1	63.8	69.7	29.3	92.7	57.1	83.7	68.4	64.7	57.3	60.1
MPF [25]	55.5	93.4	30.2	38.3	26.1	28.5	48.1	46.1	18.1	90.6	62.3	74.5	30.6	88.5	59.7	83.5	59.7	69.2	49.7	58.1
SqueezeSegV3 [23]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
KPCConv [14]	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	95.0	64.2	84.8	69.2	69.1	56.4	47.4
ACPNNet [42]	59.4	95.2	39.3	41.7	41.8	37.7	55.2	48.1	33.7	91.3	66.0	74.9	14.2	90.5	61.5	84.4	67.6	68.2	57.5	59.9
SalsaNext [24]	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
FIDNet [26]	59.5	93.9	54.7	48.9	27.6	23.9	62.3	59.8	23.7	90.6	59.1	75.8	26.7	88.9	60.5	84.5	64.4	69.0	53.3	62.8
BAAF-Net [15]	59.9	95.4	31.8	35.5	30.9	38.6	49.5	55.7	53.0	90.9	62.2	74.4	23.6	89.8	60.8	82.7	63.4	67.9	53.7	52.0
SalsaFusedNB [44]	60.2	92.7	51.2	44.6	43.9	43.2	63.5	59.2	8.1	92.3	65.9	76.9	30.5	91.2	65.7	83.0	66.0	66.5	53.5	62.9
Meta-RangSeg [27]	61.0	93.9	50.1	43.8	43.9	43.2	63.7	53.1	18.7	90.6	64.3	74.6	29.2	91.1	64.7	82.6	65.5	65.5	56.3	64.2
MSCSeg (Ours)	62.6	93.6	53.6	48.3	43.4	36.6	65.7	67.5	23.7	91.5	67.2	75.6	28.5	91.0	65.9	83.5	67.7	66.6	58.5	61.1

4 EXPERIMENT

In this section, we first introduce the experiment dataset and the details of the model implementation. Then, we discuss the performance of the proposed model in the point cloud semantic segmentation task. Finally, we provide ablation experiments to verify each module's effectiveness.

4.1 DataSet

All experiments are conducted on SemanticPOSS [4] and SemanticKITTI [3]. SemanticPOSS is a point cloud semantic segmentation dataset mainly used in autonomous driving, containing 2,988 LiDAR scans with point-wise labelling and 14 classes, a total of 6 sequences 00 to 05, of which sequence 02 is used as the test dataset. SemanticKITTI is a large-scale outdoor point cloud dataset that provides 23,201 full 3D scans for training and 20,351 for testing. Sequences 00 to 07, 09 to 10 are used as the training set, 08 as the validation set, and 11 to 21 as the test set. The dataset contains a total of 28 semantic categories, and only 19 semantic categories are used for evaluation under single-scan conditions.

4.2 Implementation details

The experiments used Pytorch 2.1.0 on two NVIDIA GeForce RTX 3090 GPUs with 24GB of memory, and recorded the inference speed of the model on a single NVIDIA GeForce RTX 2080Ti GPU. Our network used the AdamW optimizer with an initial learning rate of $2e^{-3}$. During training, we used random rotations, random point drops, and point flips for data augmentation. We trained for 50 epochs with a batch size of 4 for the segmentation experiments on the SemanticKITTI dataset and 60 epochs with a batch size of 4 for the SemanticPOSS dataset. We followed the ResBlock combination method of ResNet34 to combine BAResBlock. In the Str-Conv, we set i to 3, 5, 7. In the WinSA, we set the num head to 8 and the window size to 8. In the SCAB, we set the channel compression ratio to 8. In the ASAB, we set the size of the convolutional kernel after channel fusion to 3×3 .

4.3 Evaluation Results and Comparisons

Table 1 shows the segmentation results of various networks on the SemanticPOSS test set. We compare our method with other projection-based solutions. Our network achieves the most advanced segmentation performance, and our segmentation effect on rider and cone/stone far exceeds the performance of other models. It also shows significant improvements in other challenging categories that are typically difficult to segment accurately. These results indicate that our model has better feature extraction capabilities. Thanks to the CSCB module, we can obtain a more detailed

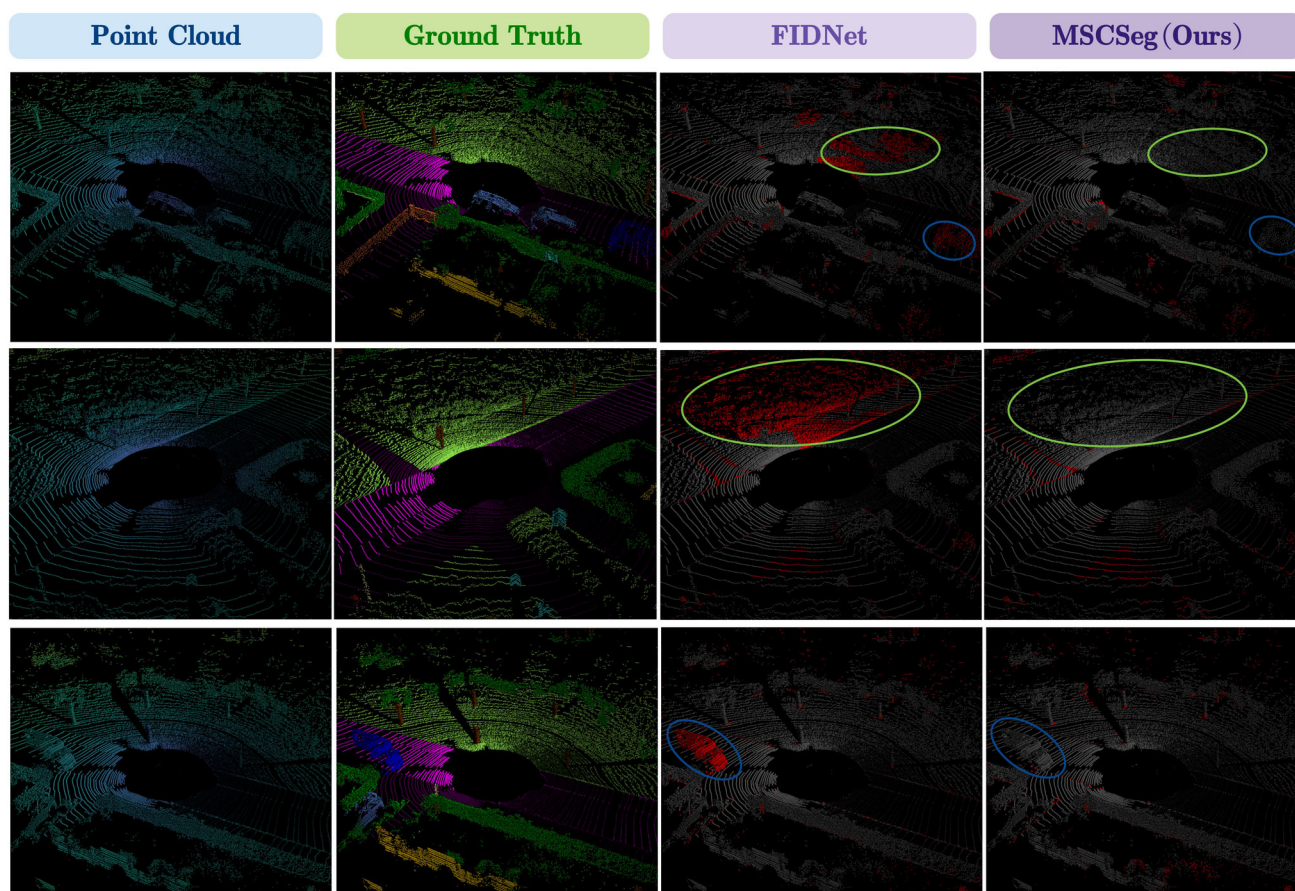


Fig. 6 Qualitative analysis on SemanticKITTI validation set. The images from left to right show the input data of the LIDAR scan frame, the segmentation ground truth, the segmentation error maps in this scan

frame for FIDNet, and the segmentation error maps in this scan frame for MSCSeg (Ours). (Red indicates the wrong predictions)

Table 3 Comparison of different convolution combinations for the BAREsBlock

Combination	Params	mIoU(%)
3×3 Conv	8.08M	61.2
$(3 \times 3)(5 \times 5)(7 \times 7)$ DWConv	7.53M	61.6
$(3 \times 5)(5 \times 7)(7 \times 9)$ DWConv	7.55M	61.9
$(5 \times 5)(7 \times 7)(9 \times 9)$ DWConv	7.57M	61.8

Table 4 Ablation study evaluated on SemanticKITTI validation set

Baseline	BAREsBlock	CSCB	FCAB	mIoU(%)
FIDNet				57.9
	✓			58.4
	✓	✓		61.3
	✓	✓	✓	61.9

feature representation of the boundary contours of small objects without treating them as part of other large categories.

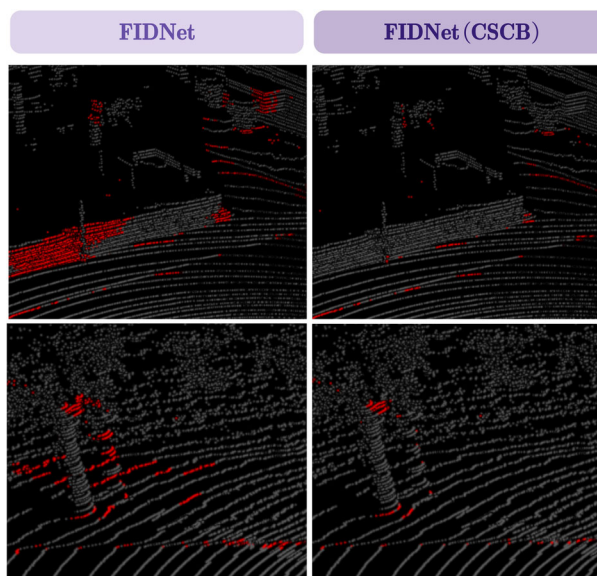


Fig. 7 Optimization effect of the CSCB.(Red indicates the wrong predictions)

We also submitted our segmentation results on the SemanticKITTI evaluation server. Table 2 provides detailed categorical quantitative results of our MSCSeg and other methods in the SemanticKITTI LiDAR semantic segmentation challenge. Our proposed method achieves better segmentation performance than most point-based and voxel-based state-of-the-art methods. MSCSeg achieves a performance gain of 3.1% on the SemanticKITTI test set compared to the baseline method FIDNet. Moreover, our projection-based model performs more efficiently than methods based on other objects, achieving an inference speed of 11.8 FPS on a single RTX 2080Ti GPU. For the SemanticKITTI dataset, which acquires point cloud data at 10 FPS, our method meets real-time requirements. These results demonstrate the effectiveness and superiority of our method.

In Fig. 6, we show the segmentation error results of our model on the SemanticKITTI 08 sequences (validation set) and compare them with the segmentation results of the baseline method FIDNet. In comparison, MSCSeg provides more accurate predictions both between and within classes, mainly due to the CSCB module.

4.4 Ablation Studies

In this section, we explore the impact of different convolution combinations on the number of parameters and model segmentation performance. We focus on 64-line vehicle-mounted LiDAR, using an input distance image size of 64×2048 . The experimental results on the SemanticKITTI validation set are shown in Table 3. When the multi-scale deep convolution combination is used to replace the traditional 3×3 convolution, not only the number of parameters is reduced, but also the mIoU of the model is generally improved. This indicates that the bigger receptive field, possessed by multi-scale convolution is beneficial to the feature learning of the method in this paper. When comparing different combinations of deep convolution, the combination of 3×5 , 5×7 and 7×9 DWConv achieves the best performance, which possesses a larger sensory field in the horizontal direction, and this asymmetric design of vertical and horizontal kernel sizes is more suitable for the feature extraction of distance images, while the combinations of 5×5 , 7×7 and 9×9 DWConv learns more redundant information in the vertical direction, resulting in inferior performance to the former.

Besides, We discuss the effectiveness of each design in the MSCSeg. Our experiments were conducted on the SemanticKITTI validation set, and the results are shown in Table 4. First, we used three standard convolutional layers to generate a 128-channel high-dimensional tensor and then replaced the original ResBlock of FIDNet with the BAResBlock, which improved the mIoU by 0.5%. Next, the mIoU was improved by 2.9% by integrating CSCB into the decoder part of the network. Fig. 7 demonstrates the optimization

effects of the proposed method after incorporating the CSCB. Compared to the segmentation error map on the left, it is evident that the CSCB reduces misclassifications at inter-class boundaries. This results in more accurate and clearer point cloud predictions at the target boundaries. To alleviate the problem of detail loss during downsampling, we proposed FCAB, which achieved a gain of 0.6% mIoU.

5 Conclusions

In this work, we propose MSCSeg, a range image-based multi-scale contextual network for point cloud semantic segmentation. MSCSeg consists of three main modules: BAResBlock, which employs depthwise convolutions with asymmetric kernel sizes to achieve cost-effective targeted feature extraction from range images; CSCB, integrating local window self-attention and multi-scale strip convolutions to capture fine-grained features within local windows while alleviating boundary ambiguity in segmentation; and FCAB, designed to hierarchically compensate for detail loss during downsampling by fusing low-level spatial details with high-level semantic features through attention mechanisms. Experiments on different outdoor street scene datasets demonstrate that MSCSeg achieves competitive performance while maintaining an inference speed that meets real-time requirements. However, as a single-frame segmentation approach, MSCSeg faces limitations in handling severe occlusion scenarios. Inspired by TS-BEV [43] and TFNet [31], Future work will explore integrating temporal-spatial feature fusion mechanisms to enhance occlusion reasoning, while maintaining computational efficiency through lightweight attention design.

Acknowledgements This work was supported in part by the National Key Research and Development Project of China (2020YFA0713503), the National Natural Science Foundation of China (61773330), the Natural Science Foundation of Hunan Province Project (2023JJ30598).

Data Availability The dataset used and analyzed in this paper is publicly available at: <http://www.poss.pku.edu.cn/semanticposs.html> and <http://semantic-kitti.org/http://semantic-kitti.org/>

Declarations

Conflict of interest The authors declare no competing interests.

References

- Li, Y., Ibanez-Guzman, J.: Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Process. Mag.* **37**(4), 50–61 (2020)
- Xiao, A., Huang, J., Liu, K., et al.: Domain adaptive LiDAR point cloud segmentation via density-aware self-training. *IEEE Trans. Intell. Transp. Syst.* (2024)

3. Behley, J., Garbade, M., Milioto, A., et al.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9297–9307 (2019)
4. Pan, Y., Gao, B., Mei, J., et al.: Semanticpos: A point cloud dataset with large quantity of dynamic instances. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 687–693. IEEE (2020)
5. Fong, W.K., Mohan, R., Hurtado, J.V., et al.: Panoptic nusenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robot. Autom. Lett.* **7**(2), 3795–3802 (2022)
6. Wang, H.F., Shan, Y.H., Hao, T., et al.: Vehicle-road environment perception under low-visibility condition based on polarization features via deep learning. *IEEE Trans. Intell. Transp. Syst.* **23**(10), 17873–17886 (2022)
7. Dewangan, D.K., Sahu, S.P.: Driving behavior analysis of intelligent vehicle system for lane detection using vision-sensor. *IEEE Sens. J.* **21**(5), 6367–6375 (2020)
8. Omeiza, D., Webb, H., Jirotko, M., et al.: Explanations in autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **23**(8), 10142–10162 (2021)
9. Klovov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 863–872 (2017)
10. Kong, L., Xu, X., Cen, J., et al.: Calib3d: Calibrating model preferences for reliable 3d scene understanding. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1965–1978. IEEE (2025)
11. Qi, C.R., Su, H., Mo, K., et al.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
12. Qi, C.R., Yi, L., Su, H., et al.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems* **30** (2017)
13. Hu, Q., Yang, B., Xie, L., et al.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108–11117 (2020)
14. Thomas, H., Qi, C.R., Deschaud, J.E., et al.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6411–6420 (2019)
15. Qiu, S., Anwar, S., Barnes, N.: Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1757–1767 (2021)
16. Zhang, F., Fang, J., Wah, B., et al.: Deep fusionnet for point cloud semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16, pp. 644–663. Springer International Publishing (2020)
17. Zhou, H., Zhu, X., Song, X., et al.: Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550* (2020)
18. Park, J., Kim, C., Kim, S., et al.: PCSCNet: Fast 3D semantic segmentation of LiDAR point cloud for autonomous car using point convolution and sparse convolution network. *Expert Syst. Appl.* **212**, 118815 (2023)
19. Wu, B., Wan, A., Yue, X., et al.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1887–1893. IEEE (2018)
20. Wu, B., Zhou, X., Zhao, S., et al.: Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 4376–4382. IEEE (2019)
21. Milioto, A., Vizzo, I., Behley, J., et al.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4213–4220. IEEE (2019)
22. Zhang, Y., Zhou, Z., David, P., et al.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9601–9610 (2020)
23. Xu, C., Wu, B., Wang, Z., et al.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16, pp. 1–19. Springer International Publishing (2020)
24. Cortinhal, T., Tzelepis, G., Aksoy, E.E.: Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In: Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15, pp. 207–222. Springer International Publishing (2020)
25. Alnaggar, Y.A., Afifi, M., Amer, K., et al.: Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1800–1809 (2021)
26. Zhao, Y., Bai, L., Huang, X.: Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4453–4458. IEEE (2021)
27. Wang, S., Zhu, J., Zhang, R.: Meta-rangeseg: Lidar sequence semantic segmentation using multiple feature aggregation. *IEEE Robot. Autom. Lett.* **7**(4), 9739–9746 (2022)
28. Cheng, H.X., Han, X.F., Xiao, G.Q.: Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In: 2022 IEEE International Conference on Multimedia and Expo (ICME), pp. 01–06. IEEE (2022)
29. Li, S., Chen, X., Liu, Y., et al.: Multi-scale interaction for real-time lidar data segmentation on an embedded platform. *IEEE Robot. Autom. Lett.* **7**(2), 738–745 (2021)
30. Ding, B.: LENet: Lightweight and efficient LiDAR semantic segmentation using multi-scale convolution attention. *arXiv preprint arXiv:2301.04275* (2023)
31. Li, R., Li, S., Chen, X., et al.: Tfnet: Exploiting temporal cues for fast and accurate lidar semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4547–4556 (2024)
32. Iandola, F.N., Han, S., Moskewicz, M.W., et al.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016)
33. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018)
34. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
35. Howard, A.G., Zhu, M., Chen, B., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017)
36. Sun, T., Zhang, Z., Tan, X., et al.: Image understands point cloud: Weakly supervised 3d semantic segmentation via association learning. *IEEE Trans. on Image Process.* (2024)
37. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
38. Mehta, S., Rastegari, M., Caspi, A., et al.: Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation.

- In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 552–568 (2018)
39. Mehta, S., Rastegari, M., Shapiro, L., et al.: Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9190–9200 (2019)
 40. Dong, X., Shi, P., Liang, T., et al.: CTAFNet: CNN–transformer adaptive feature fusion object detection algorithm for complex traffic scenarios. *Transp. Res. Rec.* **2679**(1), 1947–1965 (2025)
 41. Shi, P., Dong, X., Ge, R., et al.: Dp-M3D: Monocular 3D object detection algorithm with depth perception capability. *Knowl.-Based Syst.* **318**, 113539 (2025)
 42. Sun, X., Song, S., Miao, Z., et al.: LiDAR Point clouds Semantic Segmentation in Autonomous driving based on asymmetrical convolution. *Electronics* **12**(24), 4926 (2023)
 43. Dong, X., Shi, P., Qi, H., et al.: TS-BEV: BEV object detection algorithm based on temporal-spatial feature fusion. *Displays* **84**, 102814 (2024)
 44. Massa, K.J.L., Grobler, H.: Adapting projection-based LiDAR semantic segmentation to natural domains. *J. Vis. Commun. Image Represent.* **100**, 104111 (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.