International Journal of Computational Intelligence Systems (2024) 17:282
https://doi.org/10.1007/s44196-024-00630-5

RESEARCH ARTICLE

Channel2DTransformer: A Multi-level Features Self-attention Fusion Module for Semantic Segmentation

Weitao Liu[1] · **Junjun Wu**[1]

Received: 17 June 2024 / Accepted: 13 August 2024

Abstract
Semantic segmentation is a crucial technology for intelligent vehicles, enabling scene understanding in complex driving environments. However, complex real-world scenarios often contain diverse multi-scale objects, which bring challenges to the accurate semantic segmentation. To address this challenge, we propose a multi-level features self-attention fusion module called Channel2DTransformer. The module utilizes self-attention mechanisms to dynamically fuse multi-level features by computing self-attention weights between their channels, resulting in a consistent and comprehensive representation of scene features. We perform the module on the Cityscapes and NYUDepthV2 datasets, which contain a large number of multi-scale objects. The experimental results validate the positive contributions of the module in enhancing the semantic segmentation accuracy of multi-scale objects and improving the performance of semantic segmentation in complex scenes.

Keywords Semantic segmentation · Channel2DTransformer · Self-attention · Deep learning

1 Introduction

Semantic segmentation, as one of the core tasks in computer vision, has been widely applied in various domains, such as intelligent vehicles [1, 2], robotics [3], intelligent surveillance [4], and so on. In recent years, driven by deep learning techniques [5, 6], significant progresses have been made in semantic segmentation models in terms of accuracy and robustness. However, accurate semantic segmentation faces challenges in complex environments due to diverse object distances, resulting in varying objects' image scales. One way to improve accurate semantic segmentation is by fusing multi-level features. However, how to effectively fuse multi-level features to enhance the model's performance in segmenting multi-scale objects in complex scenes remains a challenge.

Semantic segmentation models typically adopt an encoder–decoder architecture. The encoder, such as VGG [7], ResNet [8], and others [9, 10], is responsible for extracting rich features from an input image, where the features at encoder's different levels represent the scene information from details to abstractions. Then, the decoder fuses these multi-level features to generate a final semantic mask. However, due to the substantial differences in receptive fields and semantic feature richness of multi-level features, efficiently fusing multi-level features to uncover the context information of multi-scale objects in the scene is a direction worth to exploration.

Most existing methods [11–13] focus on employing addition or concatenation operations inside various stages of decoder. These way may fail to fully exploit the inherent correlations between features at different levels, leading to features' information conflicts or losses. Attention-based fusion methods [14, 15] such as cross-attention fuse using self-attention but require flattening features into sequential tokens, leading to exponential increases in computational complexity, especially for high-resolution images. Therefore, designing a flexible module that autonomously learn multi-level features fusion strategies without adding extra computational overhead is desirable.

Inspired by the self-attention mechanism of Transformer [14], we propose an innovative multi-level features self-attention fusion module called Channel2DTransformer(C2T). Compared with the conventional Transformer, our C2T mod-

All funding mentioned in acknowledgements section is gratefully acknowledged.

✉ Junjun Wu
jjunwu@fosu.edu.cn

[1] Guangdong Provincial Key Laboratory of Industrial Intelligent Inspection Technology, Foshan University, Foshan, China

Published online: 18 November 2024

ule computes self-attention directly among two-dimensional features, ensuring that computational complexity does not exponentially grow with image resolution while autonomously capturing the local details and global information associations across different levels. The core of C2T module is to adaptively assign self-attention weights to multi-level features between their channels, and then fuse all this features by the weights, thereby achieving a more consistent and comprehensive representation of scene feature. Notably, the C2T is a plug-and-play module with good generality and scalability. The main contributions of this paper can be summarized as follows:

1. A novel Channel2DTransformer module that calculates self-attention weights between multi-level features' channels and fuses all this features by the weights is proposed.
2. We plugin our C2T module into the decoder head of two popular semantic segmentation models in comparative experiments, and the results demonstrate the effectiveness of the C2T module in improving the semantic segmentation performance and the accuracy of multi-scale objects on the Cityscapes and NYUDepthV2 datasets. We also carry out application tests and prove that the proposed module has a good generalization in our campus untrained scenes.

2 Related Works

2.1 Self-attention Mechanism

The self-attention mechanism was originally proposed to solve the sequence modeling task [16] in parallel. It dynamically assigns weights to features at various positions by calculating the correlation scores, allowing each feature to concentrate on their most relevant information. The transformer architecture proposed by Vaswani et al. [14] has achieved outstanding performance in language tasks, driving the application of self-attention mechanism in computer vision.

Vision Transformer (ViT) [17] pioneered the application to vision tasks by dividing images into patches, embedding these patches into sequences, and utilizing self-attention mechanisms to understand the relationships between patches, thereby accomplishing vision tasks. SETR [18] and TransUNet [19] improved the Transformer decoder, while SegFormer [20] innovated on the encoder. The Swin Transformer [21] was another milestone work, introducing shifted windows and hierarchical structures to achieve the state-of-the-art performance in semantic segmentation and other vision tasks. Swin v2 [22] further proposed concepts, such as post-normalization, cosine similarity, continuous relative position bias, and log-spaced coordinates, addressing challenges in model scale and adapting to different image resolutions. Additionally, methods combining CNNs and Transformers, such as PVT [23, 24], were proposed to leverage the advantages of both.

All these methods utilize self-attention to overcome CNN receptive field limitations, significantly improving multi-scale object semantic segmentation. However, most require one-dimensional sequences conversion for attention computation, increasing complexity. Our proposed module applies self-attention directly to two-dimensional feature fusion.

2.2 Multi-level Features Fusion

Multi-level features fusion strategies have always been the focus of research, especially for complex scenes understanding. Lin et al. [11] introduced the Feature Pyramid Network (FPN), which fuses multi-level features through lateral connections and upsampling operations, thereby significantly enhancing the model's capability to handle multi-scale objects detection. Semantic FPN [25] harnessed the features outputted by the FPN, subsequently processing them through a meticulously designed FPN head to perform semantic segmentation. Zhao et al. [26] designed the pyramid attention module, which computes attention weights and fuses multi-level features by attention. Qin et al. [27] extracted rich semantic information by employing a cross-attention mechanism on multi-level features. Yu et al. [28] incorporated atrous convolutions, substantially expanding the receptive field and fusing multi-scale contextual information. Zhang et al. [12] designed the ExFuse module, which recursively fuses features from lower to higher levels in a gradual manner. Wang et al. and Sun et al. [10, 29] proposed parallel high-resolution and low-resolution convolutional streams to achieve multi-level features extraction and fusion.

However, all these methods need to manually design and combine multiple components for feature fusion, and still rely on simple addition or concatenation operation inside various stages of components, lacking self-adaptability.

3 Proposed Method

The model we proposed is based on the Semantic FPN [25] framework, and then, our Channel2DTransformer module is plugged to enhance the semantic segmentation performance, as shown in Fig. 1. Our system pipeline can be divided into three parts: (1) The multi-level features are first extracted by SemanticFPN from an input image. (2) Then, our Channel2DTransformer module is employed to perform channel-wise self-attention fusion of the multi-level features. (3) Finally, the fused features are combined through addition to predict the semantic mask.

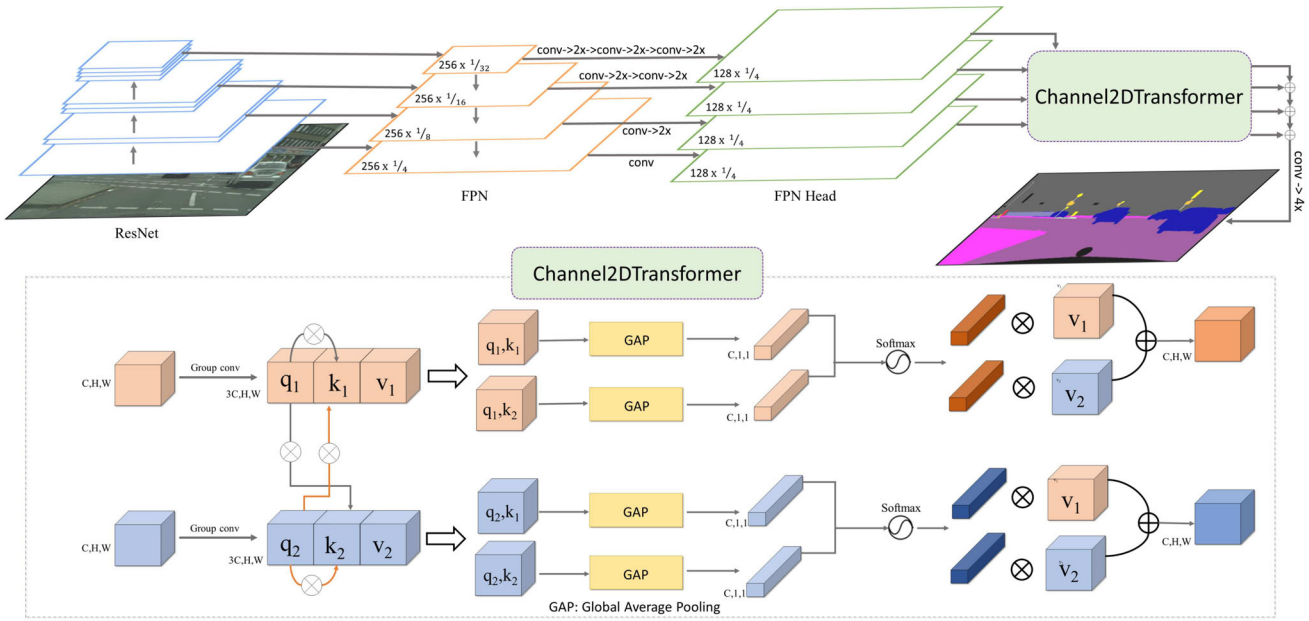

Fig. 1 Proposed Channel2DTransformer plugged into SemanticFPN for semantic segmentation

3.1 Multi-level Feature Extraction

First, we serve the output of ResNet's four stages' dense features as the multi-level features. Then, the four stages' multi-level features are passed to the Feature Pyramid Network (FPN) [11], which generates four features with the same number of channels but different sizes. The four features are subsequently input to the FPN Head, which transforms them into features with consistent size and channel numbers. This facilitates the subsequent processing and analysis by our Channel2DTransformer module. The resulting four-level features have the same size and channel numbers, but each captures different representation ranges. The higher level features represent the global information, such as object shape and semantic, while the lower level features represent the local details and textures.

3.2 Channel2DTransformer

The FPN fusion is typically performed by propagating features from higher level to lower level. However, this process may lead to the high-level information loss as the number of propagation steps increases [30], so the feature representations at each level still have differences. After that, a simple summation of all level features may cause information confusion, affecting the quality of the final representation of semantic feature. To address this, we consider the differences and correlations between the different level features. Dynamic weights are designed to assign to the fusion operation. Specially, we give higher weights to those features with similar information and lower weights to those with significant differences. This content-based weight allocation mechanism prevents excessive fusion or dilution of information, preserving valuable features and enhancing the overall performance of the model. Moreover, this strategy improves the model's adaptability to hierarchical differences, allowing fine adjustments of the contribution of each feature level during fusion.

The Channel2DTransformer module fuses multi-level features along the channel dimension and its computing multi-level channel fusion weights is similar to the self-attention mechanism in Transformer. Assuming there are n-level features to be fused, the ith level feature f_i of the cth channel is defined as f_{ic}. First, using group convolution, each f_{ic} is individually performed convolution to compute q_{ic}, k_{ic} and v_{ic}

$$q_{ic}, k_{ic}, v_{ic} = split\left(conv\left(f_{ic}\right)\right), \tag{1}$$

where the convolution operation $conv()$ produces a feature with three channels from f_{ic}. Subsequently, the feature is split using the $split()$ operation into q_{ic}, k_{ic} and v_{ic} along the channel dimension. Following the standard Transformer methodology, q_{ic} is subjected to scalar scaling:

$$q_{ic} = \frac{1}{\sqrt{H \times W}} q_{ic}, \tag{2}$$

where the symbols H and W correspond to the height and width of the feature f_i. Subsequently, f_{ic} is fused with

$\{f_{tc}\}_{t=1}^{n}$ using self-attention mechanism, resulting in the y_{ic}

$$y_{ic} = \sum_{t=1}^{n} \left(\frac{e^{GAP(q_{ic} \times k_{tc})}}{\sum_{t=1}^{n} (e^{GAP(q_{ic} \times k_{tc})})} v_{tc} \right), \tag{3}$$

where the function $GAP()$ represents the global average pooling operation. Applying Eqs. (1), (2) and (3) to all channels of all level features, we obtain $\{y_i\}_{i=1}^{n}$, which have the same dimensions and number of channels as $\{f_i\}_{i=1}^{n}$.

3.3 Semantic Prediction

After processing with the Channel2DTransformer module, the multi-level features incorporate their respective contextual information, resulting in a more comprehensive and consistent multi-level features representation. These features are then added or concatenated to form a unified feature. Finally, the unified feature's channel is adjusted using a 1x1 convolutional layer to match the number of semantic segmentation classes, and a bilinear interpolation operation is applied to obtain a semantic feature as the same size of the input image.

3.4 Training Loss

We use cross-entropy loss to compare the predicted results y_{pred} (probability distribution vectors) of each pixel with the one-hot encoded label vectors y_{true}. The computation is as follows:

$$L_{\text{pixel Loss}} = -\sum_{\text{classes}} y_{\text{true}} log(y_{\text{pred}}), \tag{4}$$

where the $classes$ represents the number of semantic categories. Finally, the total loss is defined as

$$\text{Loss} = \frac{1}{H \times W} \sum_{H \times W} L_{\text{pixelLoss}}. \tag{5}$$

3.5 Network Details

We use ResNet50 [8] as the backbone to extract image features. However, some modifications are made to satisfy the specific nature of the semantic segmentation task: that is, multiple downsampling operations can lead to the loss of fine-grained details. To avoid it, the convolutional stride in the stage 3 and stage 4 of ResNet50 is set to 1. Moreover, to prevent a decrease in receptive field size caused by the use of stride 1 convolutions, dilated convolutions with a kernel size of 3×3 are employed in the middle layer of each bottleneck in stage 3 and stage 4, with a dilation factor of 2 for stage 3 and 4 for stage 4. In the comparative experiment (Tables 1 and 2), the backbone of semanticFPN has the same modifications.

As for the Channel2DTransformer module, group convolutions with a kernel size of 1x1 are used, where the number of groups is equal to the number of feature channels. Additionally, each feature from different levels is independently processed using their own group convolution.

3.6 Training Details

First, we pretrain the modified ResNet50 model on the ImageNet [31] dataset. Then, we combine it with FPN, FPN Head, and our C2T module to train the semantic segmentation task. Training is conducted on an NVIDIA^{TM} GeForce RTX 4090 device using PyTorch. We use AdamW as the optimizer. For Cityscapes dataset, the initial learning rate is set to 1e−4, batch size is set to 5, and the training iterations are set to 160k. For NYUDepthv2 dataset, the initial learning rate is set to 2e−4, batch size is set to 14, and the training iterations are set to 160k as well. The polynomial learning rate decay strategy for both datasets during training with a decay coefficient of 0.9 is employed to facilitate model convergence.

4 Experiments

We conduct experiments on the Cityscapes outdoor and the NYUDepthV2 indoor scenes, which contain a large number of multi-scale objects. To further demonstrate the effectiveness of our C2T module, we conduct the experiment by plugging our module into the popular HRNetV2 network, which has multi-level outputs that can be easily plugged into our module.

4.1 Cityscapes Datasets

The Cityscapes dataset [32] is a collection of urban outdoor street scenes captured from the first-person perspective, containing a large number of diverse multi-scale objects (as shown in Fig. 2). It consists of 5000 high-resolution images (1024×2048 pixels) with precise pixel-level annotations for 19 semantic categories. Among the images, 2975 are used for training, 500 for validation, and 1525 for testing. During Cityscapes training, data augmentation techniques are applied, including random scaling (0.5–2.0), random cropping (769×769 pixels), and random adjustments of brightness, contrast, saturation, and hue. Following the conventional practice [25], test-time augmentation is also performed during testing.

Table 1 Comparison of different methods on Cityscapes val. set in terms of mIoU metric

Method	Backbone	#Params	GFLOPs	mIoU (%)
DeepLab [34]	D-ResNet-101	45.11M	258.74	70.42
RefineNet [35]	ResNet-101	–	–	73.67
SAC [36]	D-ResNet-101	–	–	78.16
BiSeNet [37]	ResNet-101	54.2M	145.77	78.95
PADNet [38]	D-ResNet-101	–	–	80.38
DenseASPP [9]	WDenseNet-161	39.90M	365.77	80.62
SVCNet [39]	ResNet-101	–	–	81.03
SETR [15]	ViT-Large	318.3M	1344.0	81.10
ANN [40]	D-ResNet-101	66.50M	640.28	81.37
DANet [41]	D-ResNet-101	69.70M	763.37	81.52
SemanticFPN [25]	ResNet-50	28.06M	295.71	73.10
SemanticFPN [25]	ResNet-101	47.05M	470.50	75.69
HRNetV2 [10]	HRNetV2-W48	65.86M	210.57	80.72
SemanticFPN+C2T	ResNet-50	28.06M	295.77	80.96
SemanticFPN+C2T	ResNet-101	47.05M	470.56	81.83
HRNetV2+C2T	HRNetV2-W48	65.43M	191.78	**82.08**

The GFLOPs is calculated with an image size of 768×768

Table 2 Comparison of different methods on NYUDepthV2 test set in terms of mIoU

Method	Backbone	mIoU (%)
3DGNN [42]	VGG	43.18
Kong et al. [43]	ResNet-101	44.52
LS-DeconvNet [44]	FCN	45.95
CFN [45]	FCN	47.73
ACNet [46]	ResNet-50	**48.31**
SemanticFPN [25]	ResNet-50	38.05
HRNetV2 [10]	HRNetV2-W48	46.75
SemanticFPN+C2T	ResNet-50	42.46
HRNetV2+C2T	HRNetV2-W48	47.74

The best results are highlighted in bold

4.2 NYUDepthV2 Datasets

The NYUDepthV2 dataset [33] consists of sequential frames captured from complex indoor scenes, as shown in Fig. 3. It includes 1449 RGB-D images with 40 semantic categories. Among them, 795 images are used for training, while the remaining 654 images are used for testing. Data augmentation for NYUDepthV2 follows a similar procedure as Cityscapes, with the exception that random cropping is performed with a fixed size of 480×480 pixels during each training iteration. Test-time augmentation is performed during testing as well.

4.3 Experiment on Cityscapes

The quantitative semantic performance comparisons on Cityscapes validation dataset are shown in Table 1. We divide the performance analyses into three parts. The first part (rows 1–10 in Table 1) demonstrates the performance of state-of-the-art semantic segmentation methods on Cityscapes validation dataset. The second part (rows 11–13 in Table 1) refers to two methods of SemanticFPN and HRNetV2 that produce multi-level features during the decoder stage. The third part (rows 14–16 in Table 1) involves plugging our C2T module to fuse the final multi-level features generated by the second part methods.

Table 1 reveals that plugging our C2T module into SemanticFPN has achieved remarkable performance with the ResNet50 backbone. Our SemanticFPN+C2T improves the mIoU by a significant 7.39% compared to SemanticFPN alone. The Channel2DTransformer module, based on self-attention mechanisms, selectively extracts and fuses useful features, resulting in substantial overall performance improvement.

When using ResNet101 as the backbone network, SemanticFPN+C2T outperforms DANet. The best performance is achieved by plugging C2T into HRNetV2. While the performance gain of HRNetV2+C2T over HRNetV2 is not as significant as SemanticFPN+C2T over SemanticFPN, HRNetV2's multi-branch design already captures consistent multi-level features, making even simple feature concatenation effective. Nevertheless, the plugging of our C2T module also can further enhance its performance.

Fig. 2 Example of Cityscapes sataset

Fig. 3 Example of NYUDepthV2 Dataset

It is worth noting that our C2T module adds minimal computational overhead, with 0.002M parameters and only 0.057 GFLOPs. By replacing HRNetV2's simple concatenation decoder head with our C2T module in the HRNetV2+C2T, we can reduce GFLOPs while improving performance.

The visual comparisons shown in Fig. 4 demonstrate the effective of our C2T module in enhancing multi-scale object semantic segmentation on the Cityscapes dataset.

When dealing with large-scale objects in the scene, such as the train in the first row, the fence in the second row, and the truck in the third row, the limited receptive field of CNN-based networks results in different attention ranges for different levels of features. Simply adding features from different levels will lead to feature confusion, resulting in a large-scale object segmentation has multiple category semantics. However, plugging our C2T module addresses this issue by selectively fusing features from different levels using self-attention mechanisms. Indeed, we obtain the cleaner and more accurate semantic segmentation masks for large-scale objects.

When dealing with small-scale objects, such as the motorcycle in the fourth row of Fig. 4, these two segmentation methods result in overly smooth outputs, leading to the loss of fine details. However, by plugging our C2T module, the semantic segmentation results for such small objects are refined.

Moreover, it is worth noting that when facing similar objects, such as the bicycle rider and pedestrian in the fifth row of Fig. 4, features at different levels may have ambiguity in category distinction. Simply adding features may preserve this ambiguity, making similar objects difficult to be differentiated accurately. Conversely, our Channel2DTransformer module employs self-attention fusion to assign smaller weights to ambiguous features, effectively eliminating the ambiguity and producing more consistent semantic segmentation masks.

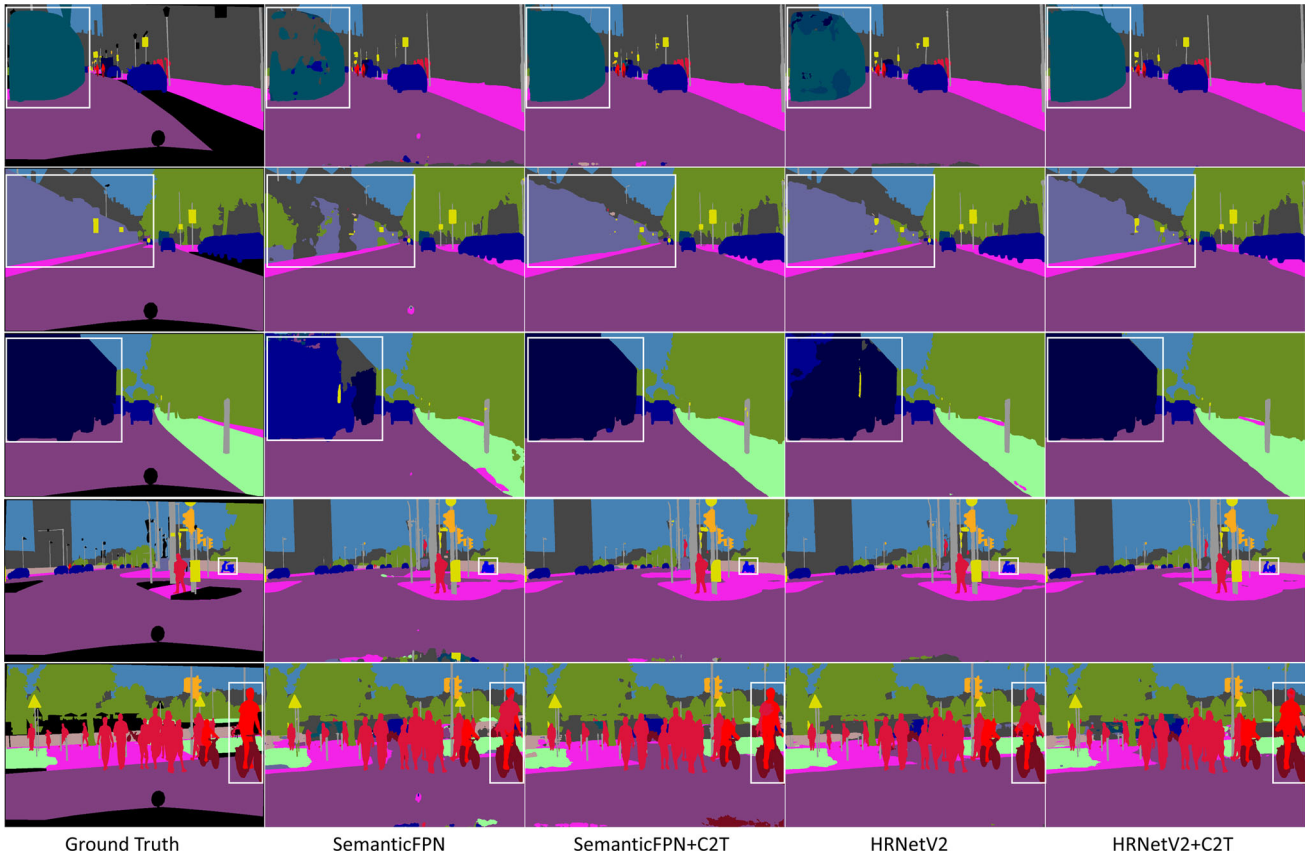

Fig. 4 Visual comparison of outdoor scenes sampled from Cityscapes

In summary, our Channel2DTransformer module, through the integration of self-attention mechanisms to fuse features across different levels, enhances the accuracy of multi-scale object segmentation in complex scenes. Furthermore, it has the potential to resolve ambiguity among similar objects, thereby providing a more powerful capability for semantic segmentation tasks.

4.4 Experiment on NYUDepthV2

Table 2 provides quantitative comparisons of various methods on the NYUDepthV2 test set. The NYUDepthV2 dataset consists of 40 categories, with complex indoor scenes and a multitude of multi-scale objects distributed in a cluttered manner, posing significant challenges for semantic segmentation tasks. Therefore, many state-of-the-art methods have low mIoU performance on this dataset.

From Table 2, it can be observed that both SemanticFPN and HRNetV2, when plugged with the C2T module, exhibit improved performance. Although HRNetV2+C2T does not outperform ACNet, it is worth noting that ACNet incorporates depth image information during the RGB processing, while our approach solely relies on RGB images for semantic segmentation. This indicates that even with single-modal input data, plugging our module can achieve comparable performance to methods that employ multimodal fusion, which is a satisfactory outcome.

The visual comparisons shown in Fig. 5 also demonstrate the ability of the Channel2DTransformer module to enhance multi-scale object semantic segmentation on the NYUDepthV2 dataset. For large-scale objects in the scene, as shown in the first to third columns of Fig. 5, plugging our C2T module not only improves the semantic segmentation results but also corrects existing segmentation errors. Similarly, for smaller objects in the scene, such as the pillows on the bed in the fourth column and the headboard in the fifth column, plugging our C2T module exhibits satisfactory improvement.

In summary, both SemanticFPN and HRNetV2, when plugged with our Channel2DTransformer module, show significant performance improvements on both the Cityscapes validation set and the NYUDepthV2 test set; these demonstrate the effectiveness of our module's feature fusion strategy based on self-attention mechanisms.

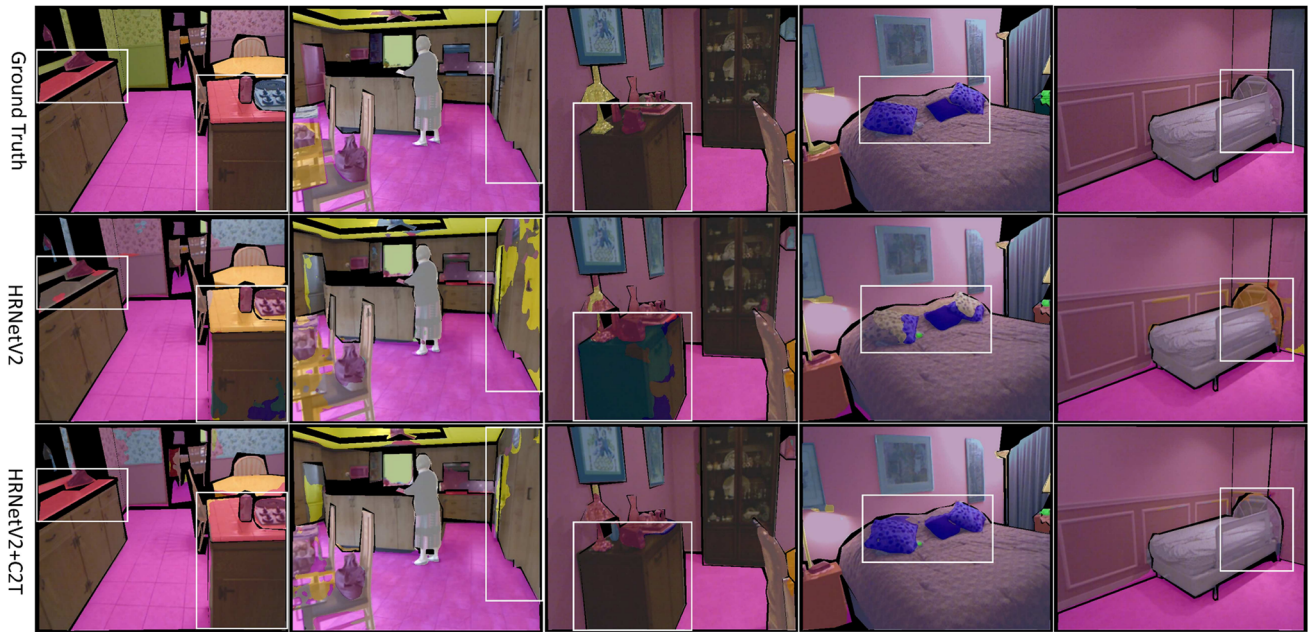

Fig. 5 Visual comparison of indoor scenes sampled from NYUDepthV2

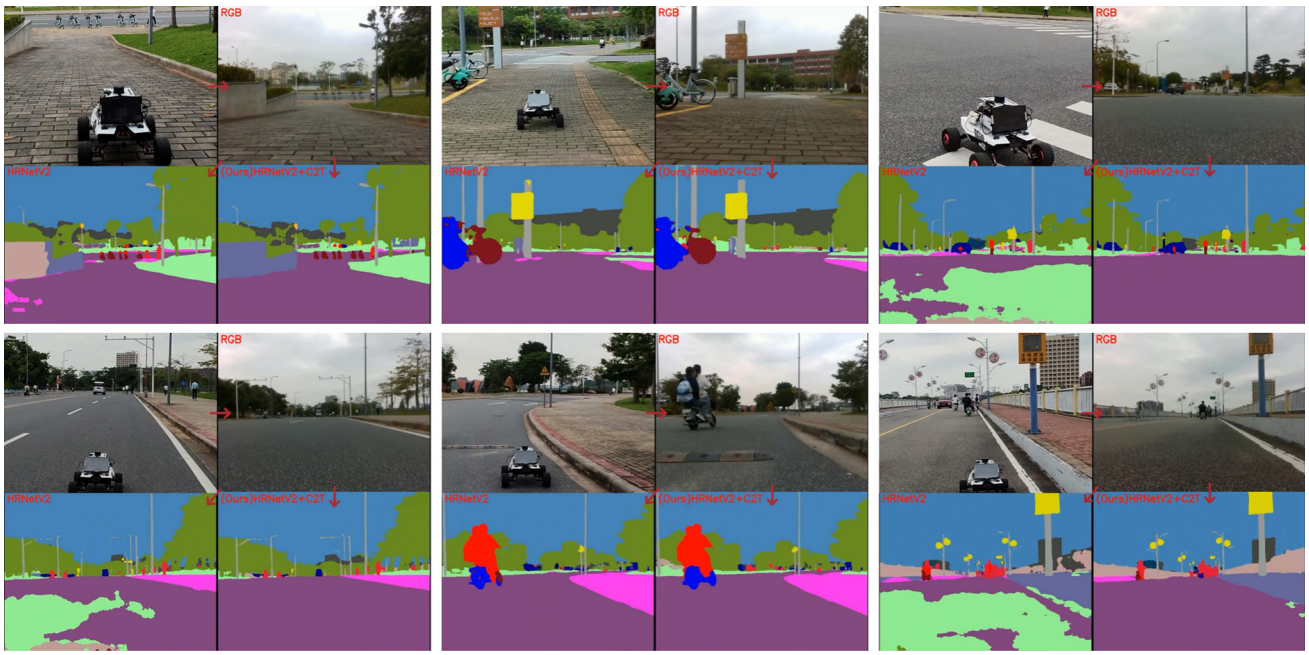

Fig. 6 Semantic segmentation comparison of campus data collected by Scout robot car

4.5 Application Tests

Generalization ability is crucial for practical applications. To verify that the model plugged our Channel2DTransformer module can achieve better semantic segmentation mask in untrained scenes, we collect a video on our campus using the Scout robot car and visually compare the semantic segmentation performance of HRNetV2 and HRNetV2+C2T, as shown in Fig. 6. Both HRNetV2 and HRNetV2+C2T have only been trained on the Cityscapes dataset.

In Fig. 6, each image block consists of four images. The upper left image in each block shows the data collection process by the Scout robot car, while the upper right image displays the campus scene captured by the camera on the Scout robot car. The lower left image shows the semantic segmentation result of the HRNetV2 method, and the lower

Table 3 Grouped convolution settings' impact on Channel2DTransformer performance

G_EqualC	False	True	False	True	True
G_KernalSize	1	1	1	3	1
G_Share	False	True	True	False	False
mIoU (%)	58.34	61.03	54.93	66.30	**68.76**

The best results are highlighted in bold

right image shows the result when our C2T module is plugged into HRNetV2.

we observe that HRNetV2 has notable semantic segmentation errors specifically for roads in our campus scene. However, when the proposed C2T module is plugged into HRNetV2, there is a noticeable improvement in the semantic segmentation of roads. Besides, the semantic masks of scene objects also have more accurate and consistent. These indicate that the Channel2DTransformer module plays a crucial auxiliary role for the network in understanding new scenes and can enhance the performance of semantic segmentation.

4.6 Ablations and Analysis

We discover that different parameter settings for the group convolution of our Channel2DTransformer module have a significant impact on the module's performance. To better understand how the module works and how to adjust the group convolution parameters to optimize the module semantic segmentation performance, we perform a series of ablation experiments (see Table 3).

The experiments are conducted on the cityscapes dataset, using semanticFPN as the base model and training for 160k iterations with the batch size of 1. "G_EqualC" indicates whether the number of groups in group convolution is equal to the number of each level feature channels. "G_KernalSize" represents the kernel size of grouped convolution. "G_Share" indicates whether different-level features share a same grouped convolution.

Comparing the last and the first columns of Table 3, with G_EqualC set from True to False, we find that when the number of groups in group convolutions is not equal to the number of feature channels, it will significantly affect the performance of the Channel2DTransformer module. This may be that each feature output by the FPN head has been already very distinguishable between its different channels. Therefore, when calculating q, k, and v, the Channel2DTransformer module should avoid destroying the channel distinguishability of each feature. Therefore, it should perform convolution operations independently on each channel instead of sharing information with other channels. This phenomenon highlights that the Channel2DTransformer module only can learn how to better fuse features from different levels but unable to learn to combine different channels of same level feature to form new representations.

Furthermore, comparing the last and the second columns, with G_Share set from False to True, we find that using separate grouped convolutions for different-level features is crucial. Different level features output by the FPN head already have their own characteristics and complexity, so they should be processed independently. These findings are further validated in the third column.

Finally, comparing the last and the fourth columns, with G_KernalSize set from 1 to 3, we find that using larger kernel in grouped convolutions proves unhelpful. This relates to the working principle of the Channel2DTransformer module that compares feature correlations of pixels in each level channel. Large kernels introduce other spatial information, damaging the original features.

4.7 Limitations

When multi-level features have already consistent, plugging the C2T module may have limited performance improvement, as seen in the HRNetV2 vs. HRNetV2+C2T comparison in Tables 1 and 2. In ablation studies Sect. 4.6, we also find that the C2T module excels at fusing features rather than learning new representations. Hence, if all the multi-level features are weakly represented, the module may not be able to enhance them.

5 Conclusion

We propose a multi-level features self-attention fusion module called Channel2DTransformer to handle the complex scenes with multi-scale objects. The module can fuse multi-level features through self-attention weights, ensuring that each level feature accurately represents targets of different scales, thereby improving the semantic segmentation performance in complex scenes. Future work will focus on applying our Channel2DTransformer module to other computer vision tasks. We believe that continuous Channel2DTransformer optimization and expansion can drive breakthroughs across fields.

Acknowledgements This work was supported in part by the National Key R&D Program of China (2022YFB4702300), in part by the National Natural Science Foundation of China (62273097), in part by the Guangdong Basic and Applied Basic Research Foundation (2022A1515140044), in part by the Research Foundation of Universities of Guangdong Province (2021KCXTD083), in part by the Foshan Key Area Technology Research Foundation (2120001011009), in part by the Guangdong Philosophy and Social Science Program (GD23XTS03), and in part by the Research project of Guangdong Special Equipment Inspection and Research Institute (2024JD-2-05).

Data Availability Code is available at: https://github.com/18128381510/Channel2DTransformer.

Declarations

Conflict of interest All authors declare that they have no conflict of interest regarding this paper.

References

1. Rizzoli, G., Barbato, F., Zanuttigh, P.: Multimodal semantic segmentation in autonomous driving: a review of current approaches and future perspectives. Technologies **10**(4), 90 (2022)
2. Xie, X., Bai, L., Huang, X.: Real-time lidar point cloud semantic segmentation for autonomous driving. Electronics **11**(1), 11 (2021)
3. Marchal, N., Moraldo, C., Blum, H., Siegwart, R., Cadena, C., Gawel, A.: Learning densities in feature space for reliable segmentation of indoor scenes. IEEE Robot. Autom. Lett. **5**(2), 1032–1038 (2020)
4. Sreenu, G., Durai, S.: Intelligent video surveillance: a review through deep learning techniques for crowd analysis. J. Big Data **6**(1), 1–27 (2019)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
6. Zhang, B., Gong, X., Wang, J., Tang, F., Zhang, K., Wu, W.: Non-stationary fuzzy neural network based on FCMnet clustering and a modified CG method with Armijo-type rule. Inf. Sci. **608**, 313–338 (2022)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
10. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3349–3364 (2020)
11. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
12. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: ExFuse: enhancing feature fusion for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 269–284 (2018)
13. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiSeNet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł, Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
15. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6881–6890 (2021)
16. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
18. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6881–6890 (2021)
19. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: TransUNet: transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
20. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: simple and efficient design for semantic segmentation with transformers. Adv. Neural Inf. Process. Syst. **34**, 12077–12090 (2021)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
22. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12009–12019 (2022)
23. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)
24. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVT v2: improved baselines with pyramid vision transformer. Comput. Vis. Media **8**(3), 415–424 (2022)
25. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6399–6408 (2019)
26. Li, H., Xiong, P., An, J., Wang, L.: Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180 (2018)
27. Qin, Z., Liu, J., Zhang, X., Tian, M., Zhou, A., Yi, S., Li, H.: Pyramid fusion transformer for semantic segmentation. arXiv preprint arXiv:2201.04019 (2022)
28. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)

29. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)
30. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
31. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009)
32. Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The ApolloScape open dataset for autonomous driving and its application. IEEE Trans. Pattern Anal. Mach. Intell. **42**(10), 2702–2719 (2019)
33. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012. Proceedings, Part V 12, pp. 746–760. Springer (2012)
34. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
35. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925–1934 (2017)
36. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2031–2039 (2017)
37. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiSeNet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)
38. Xu, D., Ouyang, W., Wang, X., Sebe, N.: PAD-Net: multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 675–684 (2018)
39. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Semantic correlation promoted shape-variant context for segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8885–8894 (2019)
40. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 593–602 (2019)
41. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
42. Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R.: 3D graph neural networks for RGBD semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5199–5208 (2017)
43. Kong, S., Fowlkes, C.C.: Recurrent scene parsing with perspective understanding in the loop. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 956–965 (2018)
44. Cheng, Y., Cai, R., Li, Z., Zhao, X., Huang, K.: Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3029–3037 (2017)
45. Lin, D., Chen, G., Cohen-Or, D., Heng, P.-A., Huang, H.: Cascaded feature network for semantic segmentation of RGB-D images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1311–1319 (2017)
46. Hu, X., Yang, K., Fei, L., Wang, K.: ACNet: attention based network to exploit complementary features for RGBD semantic segmentation. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 1440–1444. IEEE (2019)