applied sciences

MDPI

Article

A Multi-Path Semantic Segmentation Network Based on Convolutional Attention Guidance

Chenyang Feng, Shu Hu and Yi Zhang *

College of Computer Science, Sichuan University, Chengdu 610042, China; young_chenyang@163.com (C.F.); hushu@scu.edu.cn (S.H.)
* Correspondence: yi.zhang@scu.edu.cn

Abstract: Due to the efficiency of self-attention mechanisms in encoding spatial information, Transformer-based models have recently taken a dominant position among semantic segmentation methods. However, Transformer-based models have the disadvantages of requiring a large amount of computation and lacking attention to detail, so we look back to the CNN model. In this paper, we propose a multi-path semantic segmentation network with convolutional attention guidance (dubbed MCAG). It has a multi-path architecture, and feature guidance from the main path is used in other paths, which forces the model to focus on the object's boundaries and details. It also explores multi-scale convolutional features through spatial attention. Finally, it captures both local and global contexts in spatial and channel dimensions in an adaptive manner. Extensive experiments were conducted on popular benchmarks, and it was found that MCAG surpasses other SOTA methods by achieving 47.7%, 82.51% and 43.6% mIoU on ADE20K, Cityscapes and COCO-Stuff, respectively. Specifically, the experimental results prove that the proposed model has high segmentation precision for small objects, which demonstrates the effectiveness of convolutional attention mechanisms and multi-path strategies. The results show that the CNN model can achieve good segmentation effects with a lower amount of calculation.

Keywords: convolutional attention; deep learning; feature guidance; multi-path; semantic segmentation

Citation: Feng, C.; Hu, S.; Zhang, Y. A Multi-Path Semantic Segmentation Network Based on Convolutional Attention Guidance. *Appl. Sci.* **2024**, *14*, 2024. https://doi.org/10.3390/app14052024

Academic Editors: Xianghua Xie, Gary KL Tam, Frederick W. B. Li, Avishek Siris and Jianbo Jiao

Received: 1 February 2024
Revised: 22 February 2024
Accepted: 24 February 2024
Published: 29 February 2024

1. Introduction

Semantic segmentation is a long-standing topic in the computer vision domain that has attracted increasing attention in both academia and industry. Semantic segmentation models have undergone significant architectural revolutions, starting from the early convolutional neural network (CNN)-based models (e.g., FCN [1] and the DeepLab series [2–4]) to the more recently published Transformer-based methods (e.g., SETR [5] and SegFormer [6]). Compared with image classification, semantic segmentation is a dense and more precise prediction task that requires the handling of object boundaries and other details. CNN frameworks emphasize multi-scale information interaction, utilizing multi-scale contextual fusion and combining various dilated convolutions and pooling operations to aggregate multi-scale contexts. However, they cannot aggregate global information. Transformer-based models, on the other hand, address this problem effectively by splitting the input image into patches and linearly embedding them into sequences. However, they require higher computational complexity, especially when dealing with high-resolution images (e.g., remote sensing images). Moreover, Transformer-based models lack some of CNNs' inherent inductive biases, such as translation equivariance and locality, which can result in some details being ignored during global extraction.

To overcome the above-mentioned problems, we propose a novel semantic segmentation network, which incorporates a multi-path encoder–decoder structure with convolutional attention. Our model is partially inspired by SegNext [7]. Building upon the visual attention network (VAN) [8], SegNext replaces the self-attention module with a multi-scale convolutional module. To reduce the computational overhead, a simple element-wise multiplication is applied to implement spatial attention, which integrates multi-scale features.

For the decoder, features from different layers are extracted, and a Hamburger model [9] is implemented to extract the global context. In addition, a spatial and channel reconstruction module is incorporated in the main path of our model to enhance feature interaction, which also removes redundant features. Multi-path encoding allows the model to use convolutional attention to extract overall features without ignoring details and boundary information. The main contributions in this paper can be summarized as follows:

(1) A multi-path convolutional self-attention structure is proposed to enhance the learning of advanced semantic information. It also integrates global information and focuses more on the boundary information.
(2) A spatial and channel reconstruction module is developed to reinforce feature interaction, which also eliminates redundant information.
(3) Extensive experiments are conducted on mainstream datasets, where our model exhibits superior performances against other popular methods.

The rest of the paper is organized as follows: related works are discussed in Section 2. The architecture of our model is described in detail in Section 3. The experimental results an ablation studies are presented in Section 4. A final conclusion is drawn in Section 5.

2. Related Works

2.1. Semantic Segmentation

Semantic segmentation is a fundamental task in computer vision. Since the introduction of fully convolutional networks (FCNs) [1], convolutional neural networks (CNNs) [10–13] have achieved tremendous success and have become a popular architecture for semantic segmentation. Fully convolutional networks keep pushing forward this field forward via their end-to-end, per-pixel classification paradigms. They capture multi-scale features, incorporate channel attention and introduce self-attention blocks to refine contextual priors. More recently, Transformer-based methods [5,6,14–16] have demonstrated significant potential and have outperformed CNN-based approaches. The general structure of a segmentation network consists of an encoder and a decoder. ResNet [17] and DenseNet [18] are commonly adopted backbones for the encoder. Meanwhile, different decoders are advised for different emphases, including achieving multi-scale receptive fields [12], collecting multi-scale semantics [4,6,19], expanding receptive fields [2,20], enhancing edge features [11] and capturing global contexts [13,21].

2.2. Multi-Scale Blocks

Multi-scale blocks are usually employed in both the encoder and decoder [3,12]. DeepLabv3+ [4], for instance, utilizes dilated convolutions at different rates in the encoder to achieve multi-scale feature extraction. However, feature extraction at different scales lacks a well-defined fusion mechanism, often relying solely on simple concatenation. Unlike previous methods, MCAG not only captures multi-scale features in the encoder but also introduces spatial and channel reconstruction modules to better fuse features. Additionally, two branches are introduced to further integrate features at larger scales. These advancements enable our model to achieve higher performance than many existing segmentation methods.

2.3. Multi-Path Structure

Multi-path structures often appear in the encoder. MPViT [22] introduces a multi-scale embedding approach with a multi-path structure, aiming to simultaneously represent coarse and fine features for dense prediction tasks. While self-attention in Transformers can capture long-term dependencies, it overlooks structural information and local relationships within each patch. Conversely, CNNs have an obvious advantage in identifying textures over shapes during visual reasoning. Therefore, MPViT combines CNNs and Transformers in a complementary manner. However, it does not highlight the individual roles of different paths. In MCAG, the model relies on the main pathway to better learn high-level semantic

information, details and boundary information and achieves global information fusion at larger scales through other branches.

3. Method

In this section, we will describe our model in detail. The multi-path semantic segmentation network with convolutional attention guidance (MCAG) also adopts an encoder–decoder structure with a three-path layout. The main path captures the multi-scale features, which are then fused through the spatial and channel reconstruction modules. The other two paths not only learn advanced semantic information but also delineate the object boundaries. Meanwhile, the model realizes the fusion of global information at larger scales. Section 3.1 will outline the overall architecture of the encoder of MCAG. Section 3.2 focuses on the convolutional attention mechanism. Section 3.3 describes the multi-path and attention-guided fusion module. Finally, Section 3.4 describes the functionality of the decoder.

3.1. Overall Architecture

Figure 1 illustrates the architecture of MCAG. Unlike BiSeNetV2 [23] and CCNet [21] which adopt single-path architectures, we propose a three-path architecture to explore a robust semantic segmentation network with multi-path convolutional attention. Specifically, we devise a four-stage, three-path structure in which the feature maps are generated with different scales and channels starting from the second stage. Our proposed structure better utilizes global and boundary information while maintaining a lower number of parameters compared to the Transformer architecture, allowing for improved learning of high-level semantic information. The central main path (MP) serves as the primary route and incorporates the MSCAN structure. The image resolution is successively reduced to 1/4, 1/8, 1/16, and 1/32 of the original resolution across the four stages, with an increasing number of channels. Each stage is composed of a convolutional attention structure and introduces a spatial and channel reconfiguration module to enhance representative features and suppress redundant spatial features. The MP is responsible for parsing long-range dependencies, as detailed in Section 3.2.

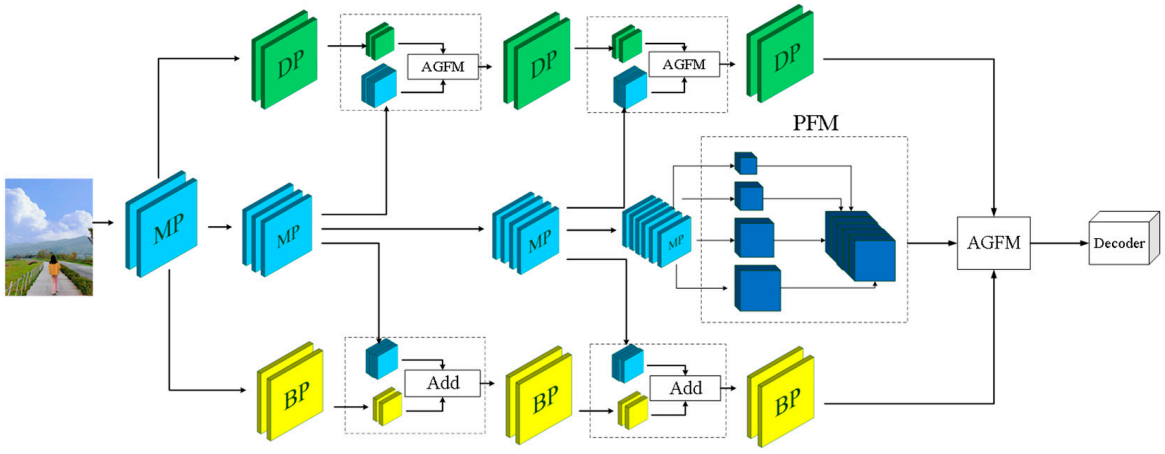

Figure 1. The encoder network architecture of MCAG. AGFM stands for attention-guided fusion module; PFM stands for pyramid pooling module; and MP, DP and BP stand for main path, detail path and boundary path respectively.

The results of the first stage of the MP are fed into two subsidiary paths: the detail path (DP) and the boundary path (BP). The DP maintains the same high resolution across all stages, emphasizing the extraction of detailed features. Starting from the second stage, guided by the MP, the DP selectively learns high-level semantic information. The BP, on the other hand, fuses with the lower-resolution MP at each stage, further integrating global information and focusing on boundary details while maintaining the same high resolution across all stages. Both paths utilize convolution-based feature extraction, with the resolution

and number of channels remaining constant, but differing in how they leverage guidance from the MP, as elaborated in Section 3.3.

After completing the fourth stage, a fusion module called the attention-guided fusion module (AGFM) is employed to merge the results from the three paths, producing the final result of the encoder section. For the decoder part, we adopt the lightweight Hamburger model [9] to generate improved segmentation results, as detailed in Section 3.4.

3.2. Convolutional Attention

We employ a convolutional attention network as the main path. For the construction of encoder blocks, a structure similar to ViT [6] is adopted. However, instead of using the self-attention mechanism, a new multi-scale convolutional attention (MSCA) module is introduced. As depicted in Figure 2b, MSCA comprises four components: depthwise convolution for aggregating local information, multi-path convolution for capturing multi-scale contexts, spatial and channel reconfiguration modules (SRU and CRU), and 1×1 convolution for modeling relationships between different channels. GELU represents the activation function, BN denotes batch normalization and Add denotes an addition operation. Here, k $\times$ k indicates the use of depthwise separable convolution with a kernel size of k $\times$ k. The outputs of the 1×1 convolution are directly used as attention weights to rebalance the input of MSCA. Mathematically, MSCA can be expressed as shown in Equations (1)–(3):

$$Att = \sum_{i=0}^{3} Conv_i(Conv_{5\times5}(F)) \tag{1}$$

$$Att\_R = Conv_{1\times1}(Att + C(S(Att))) \otimes F \tag{2}$$

$$Out = Att\_R \otimes F \tag{3}$$

where $Conv_i(i = 0, 1, 2, 3)$ represents the convolutional layers in the diagram, with kernel sizes of 1×1, 7×7, 11×11 and 21×21, respectively. C and S represent CRU and SRU, while F represents the input features. Att_R and Out denote the attention map and output, respectively. In Equation (3), the symbol $\otimes$ denotes element-wise matrix multiplication. Stacking a series of building blocks results in the proposed convolutional encoder yields MSCAN (shows in Figure 2a). MSCAN adopts a hierarchical structure, where the MP consists of four stages. Each stage consists of L MSCANs, and the spatial resolution decreases as $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, $\frac{H}{32} \times \frac{W}{32}$, where H and W are the height and width of the input image, respectively. Each stage includes a downsampling block with a batch normalization layer [24], followed by a convolutional layer with a stride of 2 and a kernel size of 3×3. The third stage contains a stack of L = 12 MSCAN modules, while the remaining stages are stacked three times each.

Additionally, spatial and channel recurrent units (SRUs and CRUs) are introduced in our MSCA module. For the SRUs, our aim is to leverage spatial redundancy in the features using a separate-and-reconstruct operation. The purpose of the separate operation is to extract information-rich feature maps from those with less information. A scaling factor from the group normalization layer is used to assess the information content of different feature maps. Specifically, given an intermediate feature map X with dimensions $N \times C \times H \times W$, where N is the batch axis, C is the channel axis and H and W are the spatial height and width axes, we firstly normalize the input as shown in Equation (4):

$$X_{out} = GN(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{4}$$

$$W_\gamma = \{w_i\} = \frac{\gamma_i}{\sum_{j=1}^{C} \gamma_j} (i, j = 1, 2, \ldots, C) \tag{5}$$

$$W = Gate(Sigmoid(W_\gamma(GN(X)))) \tag{6}$$

where μ and σ are the mean and standard deviation of X, ε is a small positive constant added for stability and γ and β are trainable affine transformations. The normalization-

related weights are obtained using the trainable parameters γ in the group normalization layer GN, as shown in Equation (5), representing the importance of different feature maps. The weighted feature map is then mapped to the (0, 1) range through a *Sigmoid* function, and a threshold gate *Gate* is applied to set weights above the threshold to 1 for informative weights W_1 and set those below the threshold to 0 for non-informative weights W_2. The entire process is represented by Equation (6).

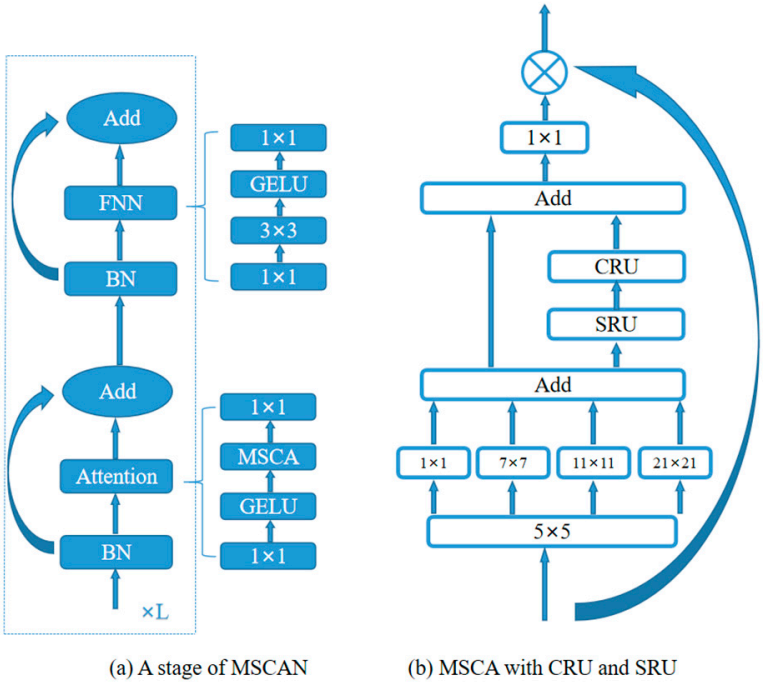

Figure 2. (**a**) The network architecture of MSCAN. (**b**) The network architecture of MSCA. CRU and SRU stand for spatial recurrent unit and channel recurrent unit, respectively.

Finally, the input features X are multiplied by W_1 and W_2 to obtain two weighted features: X_1^w, with high information content, and X_2^w, with low information content; features X_2^w with little or no spatial content information are considered as redundant. A cross operation is then applied to thoroughly combine these two differently weighted information features, enhancing the information flow between them. The cross-reconstructed features X^{w_1} and X^{w_2} are concatenated to obtain spatially refined features X^w. The entire reconstruction process is represented by Equation (7):

$$\begin{cases} X_i^w = W_i \otimes X, (i = 1, 2) \\ X_{ij}^w = Split\left(X_i^w\right), (i, j = 1, 2) \\ X_{11}^w \oplus X_{22}^w = X^{w_1}, \\ X_{21}^w \oplus X_{12}^w = X^{w_2}, \\ X^{w_1} \cup X^{w_2} = X^w. \end{cases} \tag{7}$$

where $\otimes$ denotes element-wise multiplication, Split denotes the operation of halving along the channel dimension, $\oplus$ denotes element-wise summation and $\cup$ denotes concatenation. After applying SRUs to the intermediate input features X, not only are the features with high information content separated from those with low information content, they are also reconstructed to enhance representative features and suppress redundant features in the spatial dimension.

For CRU, the aim is to exploit channel redundancy in features, utilizing a split–transform–merge strategy to further reduce redundancy along the channel dimension of spatially refined feature maps. Initially, the split operation is applied. For a given $X^w \in R^{c \times h \times w}$, its channel dimension is decomposed into αC and $(1 - \alpha)C$ components,

where $0 \leq \alpha \leq 1$, denoted as X_{up} and X_{low}, respectively. This process is expressed through Equations (8) and (9):

$$Y_1 = M^G X_{up} + M^{P_1} X_{up} \tag{8}$$

$$Y_2 = M^{P_2} X_{low} \cup X_{low} \tag{9}$$

where $M^G \in R^{\frac{\alpha C}{2r} \times k \times k \times C}$ and $M^{P_1} \in R^{\frac{\alpha C}{r} \times 1 \times 1 \times C}$ are learnable weight matrices, $X_{up} \in R^{\frac{\alpha C}{r} \times h \times w}$ and $Y_1 \in R^{C \times h \times w}$ represent the input and output from the upper path, $M^{P_2} \in R^{\frac{(1-\alpha)C}{r} \times 1 \times 1 \times (1-\frac{1-\alpha}{r})C}$ is a learnable weight matrix, $\cup$ denotes the concatenation operation, $X_{low} \in R^{\frac{(1-\alpha)C}{r} \times h \times w}$ and $Y_2 \in R^{C \times h \times w}$ are the input and output from the lower branch and r is the squeeze ratio, controlling the feature channels to balance computational costs (set to 2 in the experiments). Finally, in the fusion stage, after the transformation, there is no direct connection or addition of the two types of features. Instead, a simplified SKNet [25] method is employed to adaptively merge the output features Y_1 and Y_2 from the upper and lower branches. Initially, global average pooling is applied to gather global spatial information with channel statistics, as shown in Equation (10):

$$S_m = P(Y_m) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} Y_c(i, j) \tag{10}$$

Here, P represents the pooling operation $(m = 1, 2)$. Next, the upper and lower global channel descriptors, S_1 and S_2, are stacked together, and a channel-wise soft attention operation is applied to generate a feature importance vector, as shown in Equation (11):

$$\begin{cases} \beta_1 = \frac{e^{S_1}}{e^{S_1} + e^{S_2}}, \\ \beta_2 = \frac{e^{S_2}}{e^{S_1} + e^{S_2}}, \\ \beta_1 + \beta_2 = 1. \end{cases} \tag{11}$$

Finally, guided by the feature importance vector, the channel-refined feature Y can be obtained by merging the upper feature Y_1 and the lower feature Y_2 in a channel-wise partitioning manner, as expressed in Equation (12):

$$Y = \beta_1 Y_1 + \beta_2 Y_2 \tag{12}$$

In summary, we employ the channel refinement unit (CRU) using a split–transform–merge strategy to further reduce channel-wise redundancy in spatially refined feature maps. Additionally, CRU extracts rich representative features through lightweight convolutional operations while mitigating redundant features through simple operations and a feature reuse scheme.

In conclusion, by sequentially arranging the spatial refinement unit (SRU) and channel refinement unit (CRU), an efficient and interchangeable standard convolution operation has been established.

3.3. Multi-Path and Attention-Guided Fusion Module

The two paths of the MCAG, as illustrated in Figure 1, include the following:

The detail path (DP) is the first path, which maintains a high resolution in earlier stages and selectively learns high-level semantic information, guided by the main path. It focuses on extracting detailed features.

The boundary path (BP) is the second path, which sums with the main path at each stage, incorporating further fusion of global information and attention to boundary details. Both paths take the output from the first stage of the main path as input and consistently maintain the image resolution, aiming to fuse features at larger scales while preserving fine details.

The two paths collectively undergo three stages, each consisting of stacked modules denoted as block_DP and block_BP, which are composed of convolutions followed by batch normalization and rectified linear unit (ReLU) activation, as depicted in Figures 3 and 4.

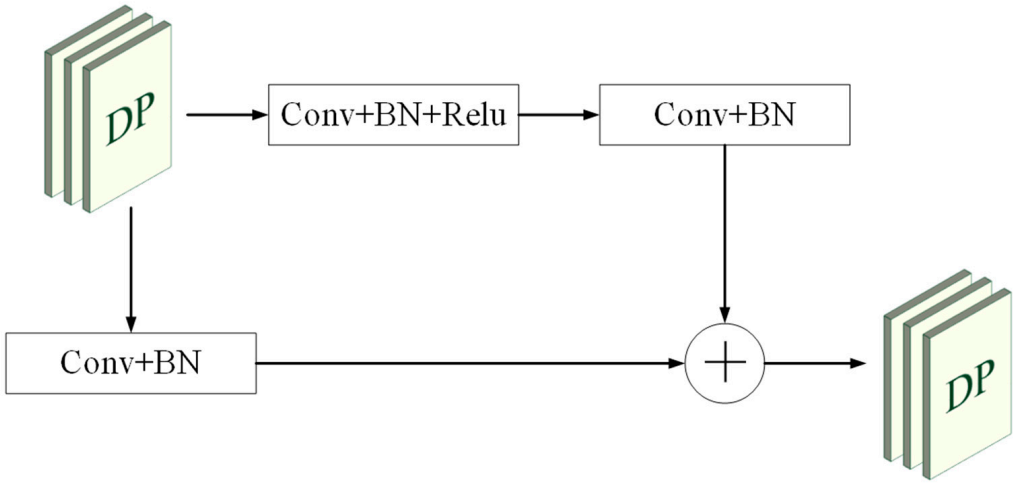

Figure 3. The structure diagram of block_DP.

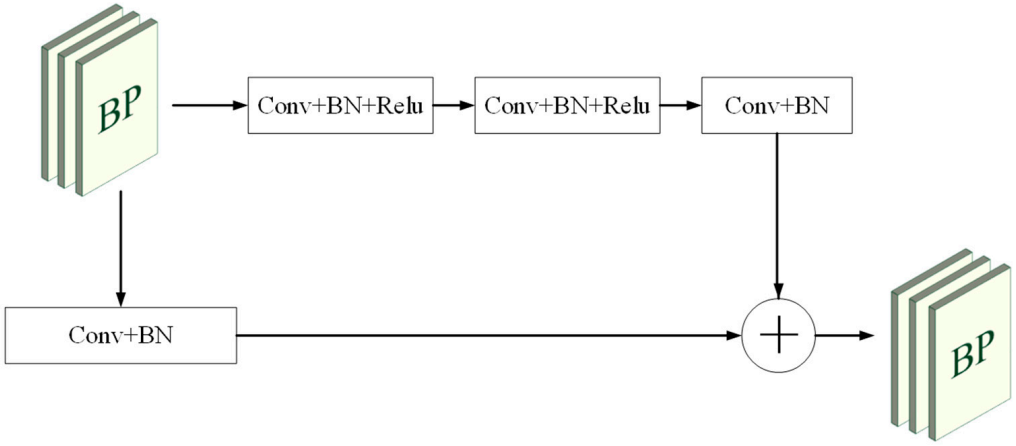

Figure 4. The structure diagram of block_BP.

Each stage of DP consists of k = 2 block_DP modules. The initial input is the output of the first stage of the main path (MP), and subsequently, the number of channels in the DP remains unchanged (except for the output channels in the fourth stage, which match those of the main path). The resolution is also consistently maintained, which better preserves the details. In each stage, the DP learns higher-level semantic information under the guidance of the main path to compensate for the loss of advanced information due to the lower number of channels and smaller convolutional kernels. In block_DP, the upper path employs a convolutional kernel with a size of 3 and a stride of 1, while the lower path uses a kernel of size 1. The primary purpose of the lower path is to adjust the number of channels (in the fourth stage) and add these to the original information from the upper path, preserving the original features as much as possible for subsequent stages. Thus, block_DP efficiently extracts features from high-resolution images with lower parameters (fewer channels and smaller convolutional kernels) and selectively learns advanced semantic information under the guidance of the main path.

Each stage of the BP consists of k = 1 block_BP module. Similar to the DP, the initial input for the BP is the output of the first stage of the main path (MP). The number of channels and resolution remain consistent in the second and third stages, and in the last stage, the number of channels matches that of the main path's fourth stage, maintaining a lower parameter count (fewer channels and smaller convolutional kernels). In contrast to

the DP, the upper path's first convolution block in block_BP doubles the feature channel count using a 1×1 convolutional kernel. The second convolutional block extracts features from the high channel count, allowing the model to better capture details and patterns in the input data, learn more types of features and improve the model's generalization ability to different samples, enhancing its robustness. The kernel size for the second block is 3×3. The third convolutional block then restores the channel count to its initial value using a 1×1 convolutional kernel. The lower path's convolutional block, similar to block_DP, primarily adjusts the number of channels (in the fourth stage) and adds the original information from the upper path. Therefore, the BP, composed of block_BP, effectively utilizes lower parameters (fewer channels and smaller convolutional kernels), relies on guidance from the main path, pays more attention to boundary information, and under the guidance of the main path, further integrates global information on a larger scale. At the end of each stage, the main path guides and provides information to both pathways.

For the DP, due to its low number of stacked convolutional layers and small kernel sizes, the main path guides its feature extraction at high resolution, allowing it to selectively learn higher-level semantics. Specifically, the main path's output from each stage starting from stage two is combined with the corresponding output of the DP and then fed into the AGFM module (attention-guided fusion module). The schematic diagram of this module is shown in Figure 5, where "dp" represents the features from the DP; "mp" represents the features from the main path; "S" represents the combination operation of *sum* and *Sigmoid*; $\otimes$ denotes element-wise multiplication, which is the weight allocation; and $\oplus$ denotes element-wise summation. The main path's high-level semantic information is selectively incorporated into the pathway, and the DP retains a significant amount of high-quality detailed information that ultimately enhances the segmentation results. The lateral connections used in [26–28] strengthen the information flow between feature maps of different scales, improving the model's representational capacity. In the AGFM, the outputs of the DP and the main path, both passed through convolutional blocks and channel expansion, are adjusted to the same resolution. Denoting these as dp and mp, the output of the *Sigmoid* function can be expressed as Equation (13):

$$S = Sigmoid(sum(dp \otimes mp)) \tag{13}$$

where the computed result S indicates the likelihood of these two pixels belonging to the same object class, *sum* represents the summation along the channel dimension and $\otimes$ denotes element-wise multiplication. When S is higher, there is reason to trust the results from the main path since it provides rich and accurate semantics, and vice versa. After obtaining S, we adjust the number of channels and resolution of the main path to match those of the DP and perform the final addition. Thus, the output of the AGFM module can be written as Equation (14):

$$Out_{AGFM} = S \otimes mp + (1 - S) \otimes dp \tag{14}$$

Therefore, in the case of deeper feature extraction, the main path can leverage higher semantic information to guide the DP in selectively learning better semantic information while preserving detailed information, ultimately optimizing the segmentation results.

For the BP, at the end of each stage, the output of the main path is directly added to the output of the BP after adjusting the number of channels and the resolution. This integrates global information and focuses on boundary information using the output features of the main path (MP).

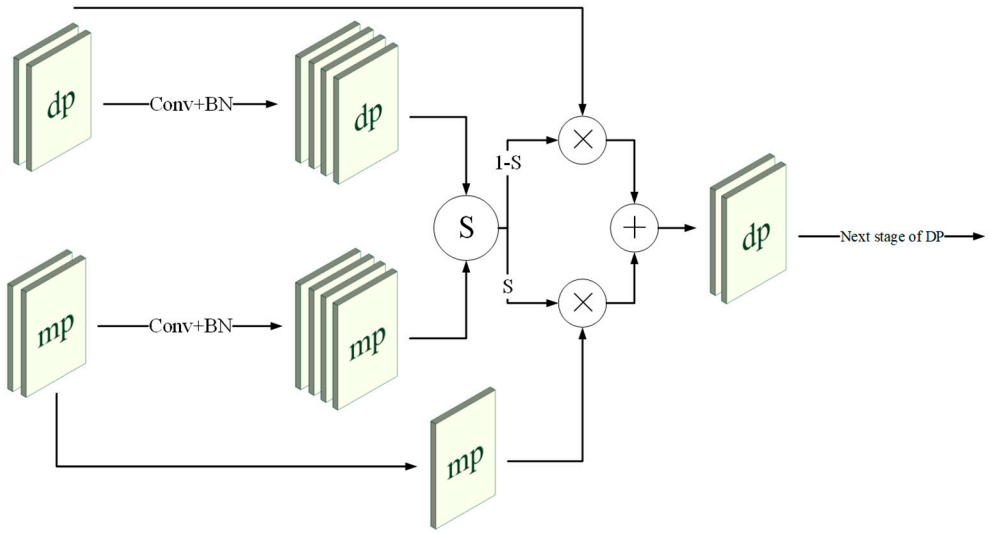

Figure 5. The structural diagram of AGFM.

To construct a better global scene prior, PSPNet introduces a pyramid pooling module (PPM), concatenating multi-scale pooled representations before the convolution layers to capture both local and global contexts. In MCAG, after the last stage of the main path, the output is fed into a parallel fusion module (PFM) to prepare for the fusion of the final three paths. This parallel fusion module enhances the context embedding capability, forming a fusion of local and global contexts to analyze global correlations. PFM processes the output of the last stage of the main path in parallel through four pooling paths, with kernel sizes of 5, 9 and 17 for the first three paths and global average pooling for the last path. It then passes through BN and ReLU layers, followed by a convolutional layer that doubles the number of channels and concatenates the results. Finally, a residual connection is established with the input features of PFM to obtain the final output of PFM, as expressed in Equations (15) and (16):

$$P = \sum_{i=0}^{3} Pooling_i(input) \tag{15}$$

$$Out_{PFM} = input + Cat(Conv(P)) \tag{16}$$

where $input$ represents the input to the PFM; $Pooling_i$ represents the four pooling paths; $Conv$ represents the combined operation of convolution, normalization and activation; and Cat denotes the concatenation operation. This output is further fused with the final results of the two paths to obtain the output of the decoder. The fusion is performed using the AGFM module, with the only difference being the fusion of three paths, as illustrated in Figure 6.

Here, dp represents the features from the DP, mp represents the features from the main path and bp represents the features from the BP. S denotes the Sigmoid operation, $\otimes$ represents element-wise multiplication, indicating weight allocation, and $\oplus$ represents element-wise summation. The BP's boundary spatial information can better optimize the contextual information from the main path and the spatial details from the DP. In this case, the BP is employed to guide the fusion of the DP and the main path. It is important to note that the main path is accurate in contextual semantics but loses too much spatial and geometric detail, especially for boundary regions and small objects. Since the BP is better at capturing boundary spatial details, it forces the model to trust the BP more concerning details and use the contextual features from the main path to fill in other regions. The computation of the AGFM in this context can be expressed as Equations (17) and (18):

$$S = Sigmoid(bp) \tag{17}$$

$$\begin{cases} i_1 = Conv((1 - S) \otimes mp), \\ i_2 = Conv(S \otimes dp), \\ Out_{AGFM} = Down(i_1 + i_2). \end{cases} \tag{18}$$

where *Down* represents the downsampled features, $\otimes$ denotes element-wise multiplication, *Conv* represents the convolution operation and mp, dp, bp represent the features from the main path, BP and DP, respectively, which are input to the AGFM.

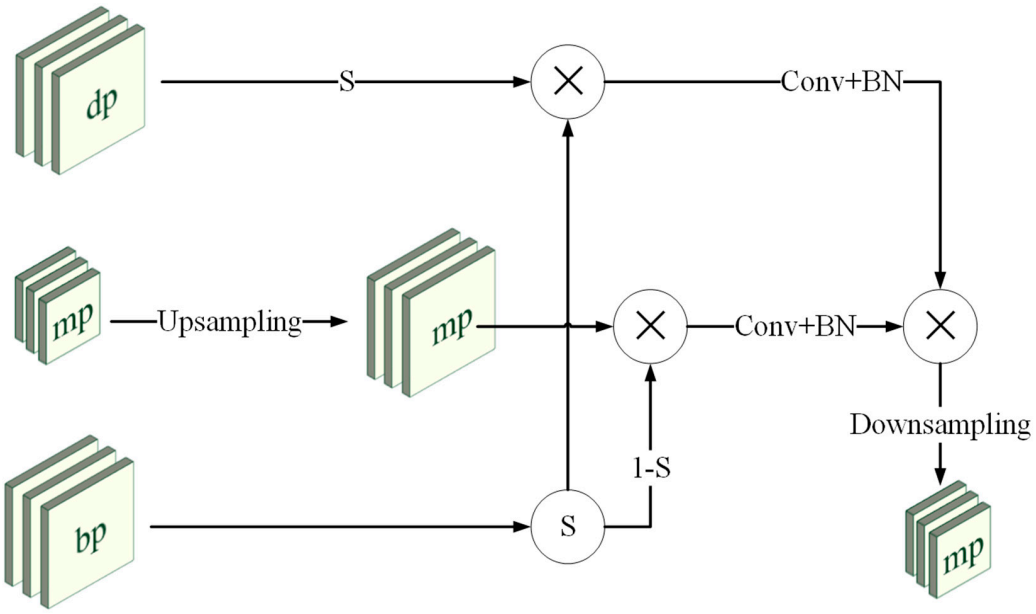

Figure 6. Fusion diagram of the main path and branch path.

3.4. Architecture of the Decoder

The encoders of previous segmentation models [5,6,20] are often pre-trained on the ImageNet dataset. To capture high-level semantics, it is common to employ a decoder on top of the encoder. This paper aggregates features from the last three stages and utilizes the lightweight Hamburger model [9] for further modeling of the global context. The Hamburger model utilizes optimization methods to solve the matrix factorization problem, decomposes the input representation into submatrices and reconstructs the low-rank embedding. When carefully handling the gradients backpropagated by matrix factorization, the Hamburger structure with different matrix factorizations performs better than self-attention in the process of modeling the global context module. Combining the powerful convolutional attention encoder from the article, the use of a lightweight decoder contributes to improved computational efficiency, as depicted in Figure 7. Here, $Stage_i$ represents the outputs from the four stages of the main path (i = 1, 2, 3, 4). The encoder part concatenates the results from the second, third and fourth stages of the main path. It further incorporates a lightweight Hamburger module for additional global context modeling.

The concatenated features then pass through a simple convolution block (convolution, normalization, activation) before being fed into a straightforward segmentation head to obtain the final decoder output. The formulation is represented as (19) and (20):

$$f = Ham(Cat(S_2, S_3, S_4)) \tag{19}$$

$$Out = MLP(Conv(f)) \tag{20}$$

where S_2, S_3, S_4 represent the outputs of the main path in the second, third and fourth stages, respectively. Here, we do not aggregate the features of stage 1 as well, because the features of stage 1 contain too much low-level information, and the information fusion of the next three stages is necessary. The operation *Cat* denotes concatenation; *Ham* represents the lightweight Hamburger module; *Conv* is a convolution block comprising convolution, normalization and

activation; and *MLP* signifies the fully connected layer of the segmentation head. With these components, we obtain the final output of MCAG.

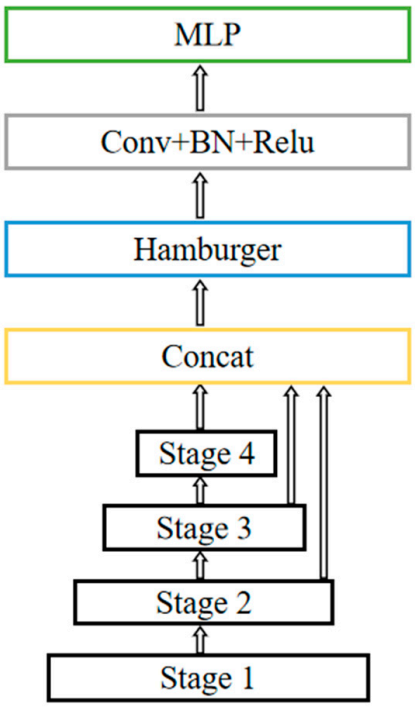

Figure 7. The decoder network architecture of MCAG.

4. Experiment

4.1. Datasets and Experimental Setup

Our network is evaluated on three popular datasets: ADE20K [29], Cityscapes [30] and COCO-Stuff [31]. ImageNet [32] stands out as the most renowned image classification dataset, featuring 1000 categories. Following a common practice in segmentation methods, this study uses ImageNet to pre-train the main path (MP) of the MCAG encoder.

ADE20K [29] is a challenging dataset with 150 semantic classes, consisting of 20,210/2000/3352 images for training, validation and testing sets, respectively. Cityscapes [30] focuses on urban scenes, presenting 5000 high-resolution images with 19 categories. The dataset is divided into 2975/500/1525 images for training, validation and testing. COCO-Stuff [31] is another challenging dataset, encompassing a total of 172 semantic classes and 164,000 images.

The experiments in this paper were conducted using PyTorch [33] and the mmsegmentation library [34]. The main route of the segmentation model's encoder was pretrained on the ImageNet-1K dataset [32]. The mean intersection over union (mIoU) was employed as the segmentation evaluation metric. All models were trained on nodes equipped with two RTX 3090 GPUs.

For the pre-training on ImageNet, the data augmentation methods and training settings were consistent with DeiT [35]. Common data augmentation techniques, including random horizontal flipping, random scaling (from 0.5 to 2) and random cropping, are applied for segmentation experiments. The batch size for the Cityscapes dataset is set to 4, while for the other datasets, it is set to 8. The AdamW optimizer [36] is used for training. The initial learning rate is set to 0.00006, and a multi-learning rate decay strategy is employed. The ADE20K model is trained for 160K iterations, and the Cityscapes and COCO-Stuff models are trained for 80K iterations.

4.2. Comparison to State-of-the-Art and Analysis

In this section, our model is compared with the state-of-the-art semantic segmentation methods, including SERT [5], SegNext [7], FLANet [37], etc., on three datasets

(ADE20K [29], Cityscapes [30], and COCO-Stuff [31]) to demonstrate the superiority of the proposed approach. The multi-scale flipping testing strategy (MS) is employed during the comparison process.

On ADE20K, we compare MCAG with state-of-the-art semantic segmentation models. As shown in Table 1, MCAG achieves a nearly 1.0% higher mIoU compared to the state-of-the-art CNN-based model SegNext-B [7], and it outperforms the fully attentional network FLANet [37] by 0.7% in mIoU. Additionally, MCAG achieves better mIoU values than the Transformer-based models MPViT [22], FASeg [38] and TSG [39] with fewer parameters. FASeg introduces a simple and effective query design for semantic segmentation called dynamic focus-aware position query (DFPQ), which dynamically generates position queries based on the cross-attention scores of the previous decoder block and the position encoding of corresponding image features. TSG, on the other hand, utilizes internal attributes of the attention map in Transformer for multi-scale feature selection in semantic segmentation. TSG introduces TSGE and TSGD in the encoder and decoder of the Transformer, respectively, to enhance the semantic segment localization performance. These results demonstrate that MCAG achieves competitive segmentation performance while introducing a multi-path self-attention mechanism at a lower computational cost than Transformer models. The asterisk (*) denotes reproduced results.

Table 1. Comparison with SOTA on ADE20K. The asterisk (*) denotes reproduced results.

Method	Params (M)	Backbone	mIoU (%)
Segformer-B0 [6]	3.8	MiT	38.0
MaskFormer [40] *	42	Swin	46.5
Segformer-B1 [6]	13.7	MiT	43.1
HRFormer-S [14]	13.5	-	45.1
HRFormer-B [14] *	56.2	-	45.9
AFFormer-base [41]	3.0	-	41.8
SegNext-B [7] *	27.6	MSCAN-B	46.72
SETR-MLA-DeiT [5]	92.59	T-Base	46.15
StructToken-PWE [42] *	38	ViT-S/16	46.6
FLANet [37]	-	HRNetW48	46.99
MPViT [22] *	52	MPViT-S	46.52
FASeg [38] *	51	R50	47.5
TSG [39]	72	Swin-T	47.5
MPCNet [43]	-	R101	38.04
MCAG (OURS)	36	MSCAN-B	47.7

On Cityscapes, we compare MCAG with state-of-the-art semantic segmentation models. As shown in Table 2, MCAG achieves a 0.51% higher mIoU compared to Segnext-B [7] on Cityscapes, surpasses FLANet [37] by 2.81% in mIoU and outperforms the Transformer-based models FASeg [38] and TSG [39] by 4.01% and 6.71% in mIoU, respectively. Moreover, MCAG achieves a 1.91% higher mIoU than PIDNet-L [44] with a lower parameter count. Additionally, StructToken [42] introduces a human-centric perspective to semantic segmentation, proposing the StructToken with structural prior (StructToken-PWE) model, which generates a coarse mask for each class based on structural priors and then progressively refines the mask. MCAG outperforms StructToken-PWE by 0.44% in mIoU. These results demonstrate that the multiple pathways in MCAG used for global information fusion and the handling of boundary information and details are highly effective. With the guidance from the main pathway, MCAG maintains excellent segmentation performance at a significantly lower computational cost than Transformer-based models. The asterisk (*) denotes reproduced results.

Table 2. Comparison with SOTA on Cityscapes. The asterisk (*) denotes reproduced results.

Method	Params (M)	Backbone	mIoU (%)
Segformer-B0 [6]	3.8	MiT	78.1
SETR [5]	311	ViT-L	79.3
MagNet [45]	-	-	67.57
HyperSeg-S [46]	10.2	EfficientNet-B1	78.1
AFFormer-base [41]	3.0	-	78.7
HRFormer-S [14]	13.5	-	81.0
SegNext-B [7] *	27.6	MSCAN-B	82.0
FLANet [37]	-	HRNetW48	79.7
FASeg [38] *	67	R50	78.5
StructToken-PWE [42] *	364	ViT-L/16	81.2
PIDNet-L [44]	36.9	-	80.6
MPCNet [43]	-	R101	78.24
LightSeg [47]	2.44	-	76.8
TSG [39]	72	Swin-T	75.8
MCAG (OURS)	36	MSCAN-B	82.51

On COCO-Stuff, as shown in Table 3, MCAG achieves a 0.9% improvement in mIoU compared with SERT. The asterisk (*) denotes reproduced results.

Table 3. Comparison with SOTA on Coco-Stuff. The asterisk (*) denotes reproduced results.

Method	Param (M)	Backbone	mIoU (%)
HRFormer-B [14] *	56.2	-	43.3
HRFormer-S [14]	13.5	-	38.9
AFFormer-base [41]	3.0	-	35.1
SegNext-B [7] *	27.6	MSCAN-B	43.5
SETR [5] *	311	ViT-L	42.7
MCAG (OURS)	36	MSCAN-B	43.6

Additionally, to highlight the good performance of MCAG concerning details and boundaries, we compare the segmentation results (mIoU %) of MCAG and SegNext on some small objects in ADE20K, including Mirror, Seat, Lamp, Box, Book, Pillow and Oven. Due to spatial limitations in this paper, the remaining results are not shown, and the results are presented in Table 4. The asterisk (*) denotes reproduced results. We believe that the semantic segmentation results for small objects can demonstrate the role of our multi-path structure to some extent. It can be observed that MCAG demonstrates its advantage in segmenting small objects. This is attributed to the guidance provided by the main route to the multiple paths in high-level semantics, as well as the emphasis on detail extraction by the two paths at high resolution.

Table 4. Comparison of specific objects with Segnext on ADE20K. The asterisk (*) denotes reproduced results.

Method	Mirror	Seat	Lamp	Box	Book	Pillow	Oven
SegNext-B [7] *	65.29	56.70	60.34	25.90	45.09	65.41	53.10
MCAG (OURS)	65.99	58.09	62.45	27.39	46.93	69.63	55.34

4.3. Visualization

In Figure 8, the visual results of MCAG on the Cityscapes dataset are presented. The first column displays the input images, the second column represents the corresponding ground truth and the third column shows the segmentation results of the MCAG method, with black rectangular regions indicating detailed displays.

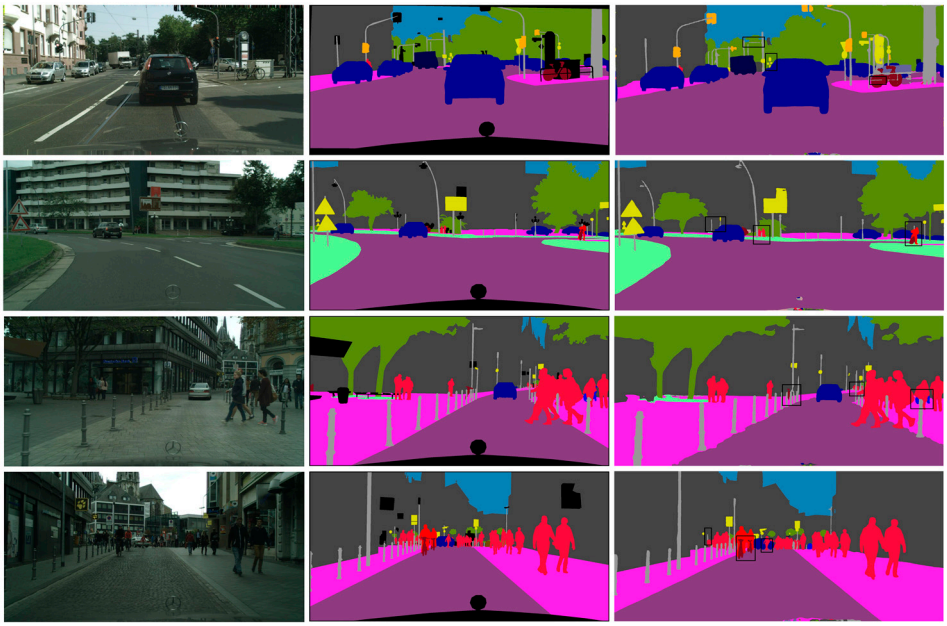

Figure 8. Visualization results of MCAG on the Cityscapes dataset. The first column displays the input images, the second column represents the corresponding ground truth and the third column shows the segmentation results of the MCAG method.

It can be observed that MCAG is more effective at identifying both boundary details and overall information. In the first set of images, MCAG successfully recognizes the railing in front of the central part of the bicycle and the seat of the bicycle, as well as pedestrians next to the utility pole. In the second set of images, above the cyclist, MCAG adeptly identifies the less noticeable lamp posts, and effectively segments the two seated individuals in the center of the image from the background bushes. In the third set of images, MCAG achieves satisfactory results in segmenting pedestrians at the end of the road, paying particular attention to details. In the fourth set of images, the model performs outstanding segmentation between the pedestrians in front of the car and the car itself, with clear distinction between the legs of the pedestrians and the road. These remarkable results stem from the robust long-range dependency-parsing ability of MCAG's main path, the subsidiary paths' exceptional focus on image details at high resolutions and the final fusion mechanism's appropriate handling of features extracted at multiple scales.

4.4. Ablation Experiment

An ablation study is conducted on the ADE20K dataset, investigating the impact of different modules on MCAG. "Multi-path" refers to the multi-path mechanism, excluding the main path; "Attention-guided" indicates the utilization of the AGFM (attention-guided fusion module); and "Parallel Aggregation" signifies the use of the PFM (parallel fusion module). The results are presented in Table 5.

Table 5. Ablation experiments of each module. ✓ indicates that the module is used.

Multiple Paths	Attention Guidance	Parallel Aggregation	mIoU (%)
✓	✓	✓	47.70
✓	✓		47.55
		✓	46.94
✓		✓	47.30
			46.72

Among the modules, "Multiple Paths without Attention Guidance" represents that the guidance of the main route to the DP and the guidance to the BP are simply added together. It can be observed that each component contributes to the model's final performance. When using both multiple paths and attention guidance, the mIoU is 0.84% higher than having only one main route. If there is no attention guidance mechanism and a simple addition of main route and branch results is performed, the result is lower by 0.4%. These two findings indicate that the proposed multi-path and attention guidance from the main route are both effective and necessary.

5. Discussion and Conclusions

In this paper, we propose a multi-path semantic segmentation network with convolutional attention guidance (dubbed MCAG). It has a multi-path architecture and feature guidance, which forces the model to focus on the object boundaries and details. It also explores multi-scale convolutional features through spatial attention, and captures both local and global contexts in spatial and channel dimensions in an adaptive manner.

The results of the ablation experiments show that the role of each module is indispensable and that convolutional attention provides powerful global feature extraction capabilities with low computational complexity. In the traditional convolution model, the details of small objects and the features of boundaries are easily lost during multi-layer convolutional extraction, and they are difficult to recover. So, the multiple paths are particularly important for the feature extraction of details and boundaries, which can be reflected in Table 4 showing the semantic segmentation results of small objects, thanks to the extraction ability of local features of our multi-path approach. Finally, a good fusion mechanism is also necessary. This paper does not use the traditional simple fusion mechanism; a PFM can better fuse global and local information. The experimental results indicate that MCAG surpasses the performance of state-of-the-art Transformer-based methods to a certain extent with a lower parameter count. The study suggests that CNN-based approaches can still outperform Transformer-based methods. It is hoped that this paper will encourage researchers to further explore the potential of CNNs.

Author Contributions: Conceptualization, C.F.; Investigation, C.F.; Methodology, C.F.; Project administration, S.H. and Y.Z.; Resources, Y.Z.; Software, C.F.; Supervision, S.H. and Y.Z.; Validation, C.F.; Visualization, C.F.; Writing—original draft, C.F.; Writing—review and editing, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
2. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
3. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
4. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
5. 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 equence-to-sequence perspective with Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6881–6890.

6. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. Segformer: Simple and effificient design for semantic segmentation with Transformers. *Adv. Neural Inform. Process. Syst.* **2021**, *34*, 12077–12090.
7. Guo, M.H.; Lu, C.Z.; Hou, Q.; Liu, Z.; Cheng, M.M.; Hu, S.M. Segnext: Rethinking convolutional attention design for semantic segmentation. *Adv. Neural Inform. Process. Syst.* **2022**, *35*, 1140–1156.
8. Guo, M.H.; Lu, C.Z.; Liu, Z.N.; Cheng, M.M.; Hu, S.M. Visual attention network. *arXiv* **2022**, arXiv:2202.09741. [CrossRef]
9. Geng, Z.; Guo, M.H.; Chen, H.; Li, X.; Wei, K.; Lin, Z. Is attention better than matrix decomposition? In Proceedings of the 2021 International Conference on Learning Representations, Virtual, 3–7 May 2021.
10. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
11. Bertasius, G.; Shi, J.; Torresani, L. Semantic segmentation with boundary neural fifields. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3602–3610.
12. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
13. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef] [PubMed]
14. Yuan, Y.; Fu, R.; Huang, L.; Lin, W.; Zhang, C.; Chen, X.; Wang, J. Hrformer: High-resolution vision Transformer for dense predict. *Adv. Neural Inform. Process. Syst.* **2021**, *34*, 7281–7293.
15. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 7262–7272.
16. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision Transformers for dense prediction. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 12179–12188.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
20. 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.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
21. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 603–612.
22. Lee, Y.; Kim, J.; Willette, J.; Huang, S.J. Mpvit: Multi-path vision Transformer for dense predtion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7287–7296.
23. Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; Wei, X. Rethinking bisenet for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9716–9725.
24. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; pp. 448–456.
25. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 5, pp. 510–519.
26. Hong, Y.; Pan, H.; Sun, W.; Jia, Y. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv* **2021**, arXiv:2101.06085.
27. Orsic, M.; Kreso, I.; Bevandic, P.; Šegvić, S. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12607–12616.
28. 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.* **2020**, *43*, 3349–3364. [CrossRef] [PubMed]
29. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ade20k dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 633–641.
30. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
31. Caesar, H.; Uijlings, J.; Ferrari, V. Coco-stuff: Thing and stuff classes in context. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1209–1218.
32. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.

33. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inform. Process. Syst.* **2019**, *32*, 8026.
34. Contributors, M. MMSegmentation: Openmmlab Semantic seg213mentation Toolbox and Benchmark. 2020. Available online: https://github.com/open-mmlab/mmsegmentation (accessed on 1 July 2022).
35. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning (ICML 2021), Online, 18–24 July 2021; pp. 10347–10357.
36. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
37. Song, Q.; Li, J.; Li, C.; Guo, H.; Huang, R. Fully attentional network for semantic segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; pp. 2280–2288.
38. He, H.; Cai, J.; Pan, Z.; Liu, J.; Zhang, J.; Tao, D.; Zhuang, B. Dynamic Focus-aware Positional Queries for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 11299–11308.
39. Shi, H.; Hayat, M.; Cai, J. Transformer scale gate for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 3051–3060.
40. Cheng, B.; Schwing, A.G.; Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. In Proceedings of the NeurIPS 2021, Online, 6–14 December 2021.
41. Dong, B.; Wang, P.; Wang, F. Head-free lightweight semantic segmentation with linear transformer. *arXiv* **2023**, arXiv:2301.04648. [CrossRef]
42. Lin, F.; Liang, Z.; Wu, S.; He, J.; Chen, K.; Tian, S. Structtoken: Rethinking semantic segmentation with structural prior. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 5655–5663. [CrossRef]
43. Liu, Q.; Dong, Y.; Jiang, Z.; Pei, Y.; Zheng, B.; Zheng, L.; Fu, Z. Multi-Pooling Context Network for Image Semantic Segmentation. *Remote Sens.* **2023**, *15*, 2800. [CrossRef]
44. Xu, J.; Xiong, Z.; Bhattacharyya, S.P. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 19529–19539.
45. Huynh, C.; Tran, A.T.; Luu, K.; Hoai, M. Progressive semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16755–16764.
46. Nirkin, Y.; Wolf, L.; Hassner, T. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4061–4070.
47. Lei, X.; Liang, J.; Gong, Z.; Jiang, Z. LightSeg: Local Spatial Perception Convolution for Real-Time Semantic Segmentation. *Appl. Sci.* **2023**, *13*, 8130. [CrossRef]