

Article

Reconsidering Multi-Branch Aggregation for Semantic Segmentation

Pengjie Cai , Derong Yang, Yonglin Zou, Ruihan Chen  and Ming Dai *

School of Mathematics and Computer, Guangdong Ocean University, Zhanjiang 524088, China; caipengjie@stu.gdou.edu.cn (P.C.); yangderong1@stu.gdou.edu.cn (D.Y.); zouyonglin@stu.gdou.edu.cn (Y.Z.); 202111621104@stu.gdou.edu.cn (R.C.)

* Correspondence: daiming_dm@foxmail.com

Abstract: For semantic segmentation tasks, there are problems in using multi-branch structures to enrich feature maps and aggregate different branches of feature maps in a certain network depth, such as the insufficient richness of feature maps and the incomplete aggregation of feature maps. Given multi-branch feature maps and branch aggregation, this study proposes a lightweight method, called multi-branch aggregation atrous spatial pyramidal pooling, by introducing an attention mechanism and CARAFE to enrich the features of the feature maps, giving the feature maps adaptive parameters, periodically adjusting the adaptive parameters, and aggregating the feature maps in both the vertical and horizontal directions. First, the atrous pyramid is retained and the attention mechanism and CARAFE are used to handle the pooling features to obtain 10 different feature maps. Secondly, giving each feature map a cascaded adaptive parameter and periodically adjusting the adaptive parameter to promote or suppress certain feature maps prevents the model from being at a local minimum or saddle point for long periods due to aleatory uncertainty. Finally, the feature maps are vertically aggregated, horizontally aggregated, and weighted for summation. This work demonstrates a competitive performance on the benchmark dataset, with an improvement of 1.88% in MPA, 0.66% in FWIoU, and 1.29% in MIoU compared to atrous spatial pyramid pooling. The benchmark datasets include PASCAL VOC 2012 and CIFAR-100.

Keywords: semantic segmentation; multi-branching; adaptive parameters; branch aggregation



Citation: Cai, P.; Yang, D.; Zou, Y.; Chen, R.; Dai, M. Reconsidering Multi-Branch Aggregation for Semantic Segmentation. *Electronics* **2023**, *12*, 3322. <https://doi.org/10.3390/electronics12153322>

Received: 3 July 2023

Revised: 31 July 2023

Accepted: 1 August 2023

Published: 3 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Background

Semantic segmentation is a typical computer vision (CV) problem that involves taking some raw data as input and converting them into a mask with regions of interest, enabling pixel-level classification of images. Semantic segmentation is a downstream task of computer vision and provides services for other tasks in computer vision, such as target detection [1,2] and pose estimation [3], which can enhance the task and improve recognition accuracy. In real life, semantic segmentation has a wide range of promising applications, such as medical image segmentation [4,5], instance segmentation [6], depth estimation [7], and other fields.

With the great success of deep learning in the field of computer vision, semantic segmentation based on deep learning has also become the mainstream direction of research nowadays, such as the earliest FCN (full convolution network) [8], the U-net series [4,5], the deeplab series [9,10], the HrNet series [6,11], etc.

HrNetV2 [11] uses a complex topology to aggregate interleaved group convolution blocks [12] with different resolutions, which improves the MIoU (Mean Intersection over Union) and also increases the training cost. U-net++ [4,5] uses a complex network structure design and uses deep supervision [13,14] to aggregate all sub-networks, which accelerates the convergence of sub-networks. For HRNetV2 [6,11], U-net++ [4,5] builds on HrNet and U-net, respectively, making the network structure more complex while increasing

the model size by more than a few fold. Lightweight network structure designs such as Deeplab [9,10] and FastFCN [15], which use FPN (Feature Pyramid Network) [16] techniques to cascade different feature maps, can also give good results. However, by cascading dilation convolution [17] and pooling convolution [18], Deeplab [9,10] obtains atrous spatial pyramid pooling (ASPP) with different sensory fields, whose feature maps are not rich enough and feature map aggregation is not complete. This research aims to design a model with a simple network topology, rich in feature maps and thorough in feature map aggregation.

Therefore, this study improves ASPP through a network structure design approach along with a lightweight design. Thus, ASPP is improved based on the network structure design approach to make it more accurate.

Existing network structure design methods, such as cascading [10,16,19], pruning [4,5], and weighted summation [20], are available. Among them, only cascading is utilized in ASPP and does not use weighted summation. Therefore, this study proposes a weighting method called cascaded adaptive parameters (CAPs), for which atrous spatial pyramid pooling of multi-branch aggregation, MBA-ASPP, is one.

MBA-ASPP based on multi-branch aggregation is divided into three modules, namely multibranch atrous spatial pyramid pooling (MASPP), cascaded adaptive parameters (CAPs), and Overlap MobileNetV2 Residual Block (OMRB).

Firstly, for Module 1, the feature map of ASPP is enriched by adding pooling branches, attention mechanisms [21,22], and CARAFE (Content-Aware Reassembly of Features) [23], while introducing more local or global information.

Secondly, for Module 2, the main design is a weighting method that has both adaptive weighted summation and cascaded feature dimensions, thus using cascaded adaptive parameters. Due to the excessive number of branches, CPAs can lead to positive and negative feedback of multi-branch feature maps subject to change uncertainty, so a period coefficient strategy is used to initialize the CAPs for a specific period.

Finally, for Module 3, the main design is to design a strategy that can aggregate multiple branches, use multi-branch features for vertical aggregation, and use a horizontal aggregation strategy using MobileNetV2 Residual Blocks (MRB) [24] to weight and sum the features obtained from vertical and horizontal aggregation to obtain a more accurate segmentation result.

In summary, this study provides an improved branching and aggregated multi-branching strategy for ASPP [9,10]. It is more accurate than ASPP.

2. Our Contributions

The contributions of this study are as follows:

- (1). A multibranch atrous spatial pyramid pooling module is proposed that preserves the atrous pyramid, uses the attention mechanism and CARAFE to handle global pooling features, extends the number of branches to 10, and enriches the feature map.
- (2). We propose a cascaded adaptive parameters (CAPs) module that adaptively promotes or suppresses each branch weight ratio and adjusts the weight ratio using a periodic coefficient strategy (PCS) to avoid models being at local minima or saddle points for long periods due to chance uncertainty, while preserving the width of the cascaded feature dimension and featuring weighted summation.
- (3). An Overlapping MobileNetV2 Residual Blocks module is proposed to improve the aggregation of different branches using multi-branch features for vertical aggregation, the MobileNetV2 Residual Block module for their horizontal aggregation, and weighting summation, which ensures the aggregation of multiple branches.

3. Related Work

3.1. Image Segmentation

Since the introduction of FCNs [8], research scholars have proposed many advanced techniques to improve FCN structured networks, and many impressive network mod-

els have evolved in the field of image segmentation. For example, U-net [4,5] uses an encoder–decoder [25] network structure to achieve good performance in medical image segmentation; U-net++ [4,5] overlays several U-net sub-networks of different sizes on top of U-net, then uses deep supervision [13] to weight and sum the output layers of all U-net sub-networks to achieve accelerated convergence of U-net sub-networks, and finally uses pruning to speed up the model prediction. HrNet [6,11] connects high-to-low resolution sub-networks in parallel. Unlike most image segmentation networks that use low-level and high-level fusion methods, Hrnetv1's repetitive multi-scale fusion, which uses the same depth and similar level of low-resolution upsampling to fuse high-resolution and high-resolution downsampling to fuse low-resolution, is useful in semantics. In the output layer of the classifier, U-net++ uses deep supervision to aggregate different sub-networks through the loss of different sub-networks, while Hrnet aggregates features of different resolutions through complex topology maps. Both U-net++ and Hrnet use complex methods to aggregate features from different branches, but this study uses a simpler aggregation method.

3.2. Pyramid Pooling

Since Zhao Hengshuang et al. [19] proposed PSPNet through the pyramid pooling module (PPM), which aggregates contextual information from different regions and mines the ability of global contextual information, solving the lack of suitable strategies to exploit category information in global scenes for networks based on the FCN model [8], PSPNet [19] has obtained global image-level features for scene parsing. Semantic segmentation opens up a completely new tool. Based on PPM [19], many excellent network models have evolved, such as Deeplabv2 [9] and Deeplabv3Plus [10] using ASPP, and FastFCN [15]. PSPNet, Deeplab, FastFCN, etc., PPMs, which use multiple different CNN layers by cascading pooling layers, obtain global image-level features for semantic segmentation problems by providing a suitable strategy to exploit global contextual information. They are also widely used in other fields such as scene analysis, target detection [26], etc. The global image-level features captured by PPMs are not rich enough, so SPPNet [27] captures richer global image-level features through spatial pyramid pooling (SPP), which uses multiple pooling layers to downsample and cascade to obtain a vector with global image-level features to improve the accuracy of the model's spatial layout and object deformation. This study also uses multiple pooling layers to capture global contextual information and obtains a matrix, not a vector. In recent years, the pyramidal pooling transformer network (P2T) [28] has been proposed, which is a pyramidal pooling network based on a multi-headed attention mechanism. P2T [28] differs from PVT [29] in that it uses multiple pooling operations to extract pooling features, while P2T applies the idea of pyramidal pooling to the visual transformer to reduce the length of sequences and learns contextual features more efficiently. Therefore, this work also invokes the Pyramidal Pooling Transformer (PPT, the base unit of P2T) [28] to build the network. The transformer is a model that has an attention mechanism to speed-up model training.

3.3. Small Sample Image Classification

In many scenarios, the number of samples in the dataset is insufficient, or the data samples are difficult to collect, resulting in insufficient data to train the model. Thus, small-sample image classification [20] was created. An AFP module (Adaptive Feature Processing module) is a multiplexed feature processing module which, unlike most neural networks, adopts the approach of widening the network. This is in contrast to the AFP module (Adaptive Feature Processing module) [20], which is a multiplexed feature processing module which differs from most neural networks by adopting the method of widening the network instead of deepening it, thus obtaining a rich feature map in a shallower layer. In this study, adaptive parameters are also used, so that different branches can learn their weight shares adaptively, except that MBA-ASPP retains the dimensionality of the

cascaded feature map and the number of branches at 10, while the AFP module reduces the dimensionality of the feature map and the number of branches to 4.

3.4. Person Re-Identification

Person re-identification is a computer vision task in which the goal is to match a person's identity across different cameras or locations in a video or image sequence. OSN (Omni Scale Net) [30,31] uses full-scale features (capturing features at different spatial scales and referring to features at isomorphic and heteromorphic scales as full-scale features) and uses a through-unity aggregation gate (assigning a weight factor to the features at each scale and summing the weights). OSN [30,31] tends to detect the adequate local scale for the segmentation via computing various dilations and their multi-branch aggregation.

4. Model Approach

In this section, the general idea of the MBA-ASPP is first presented, followed by a demonstration of the MBA-ASPP network architecture. Finally, some network implementation details are presented.

The structure of atrous spatial pyramid pooling of multi-branch aggregation (MBA-ASPP) is shown in Figure 1. Using the features extracted from ResNet [32] as inputs, as in ASPP [10], the input is extracted to different branches using different branching operations, but MBA-ASPP has twice as many branches as ASPP. MBA-ASPP assigns a CAP to each branch while using a periodic coefficient strategy (PCS) and finally, all the branches are cascaded into the OMRB module.

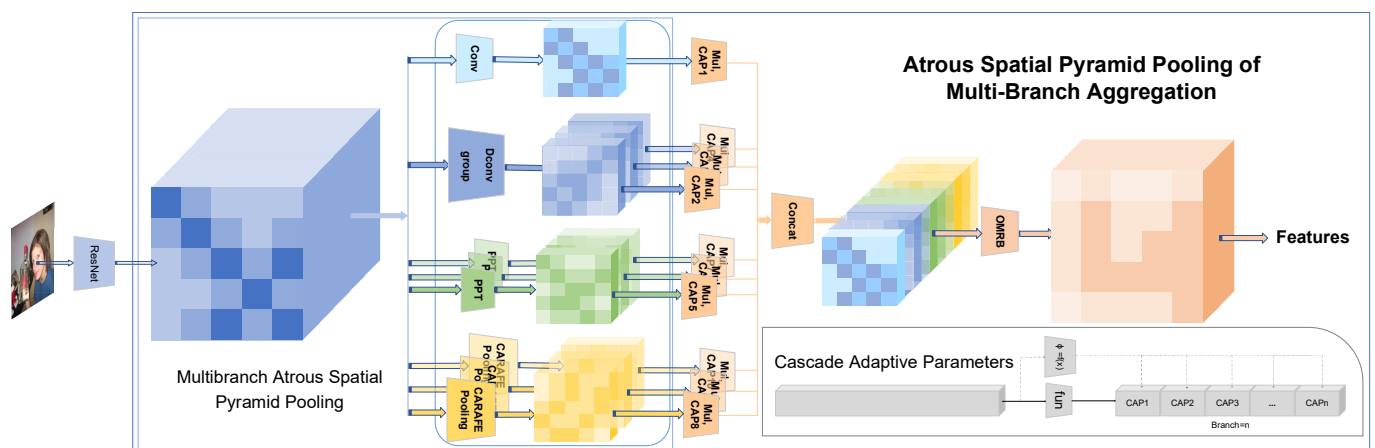


Figure 1. MBA-ASPP structure diagram.

4.1. Multibranch Atrous Spatial Pyramid Pooling (MASPP)

MASPP is designed based on ASPP [10], which uses three different types of branches with the structure shown in Figure 2, namely a 1×1 convolutional layer, a dilated convolutional group, and a global pooling layer.

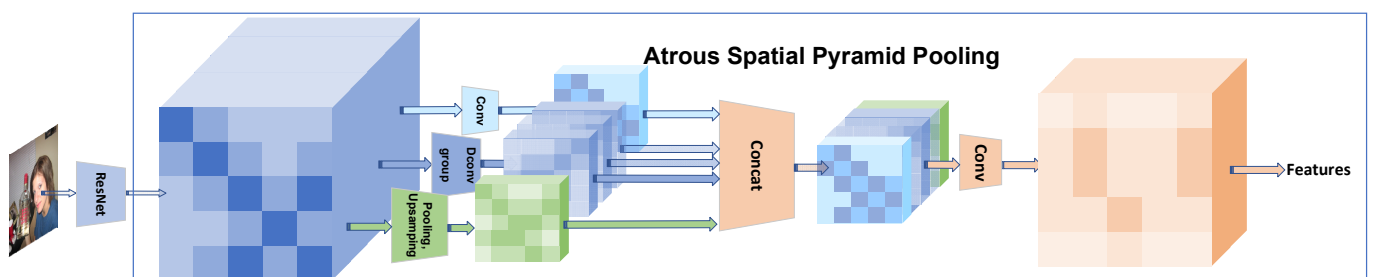


Figure 2. ASPP structure diagram, where D denotes the dilation factor.

MASPP consists of a 1×1 convolutional layer, a dilated convolutional group, three CARAFE poolings, and three PPTs, with the structures shown in Figures 3–5. The details are shown below:

$$P_1 = \text{conv}_{1 \times 1}(X) \quad (1)$$

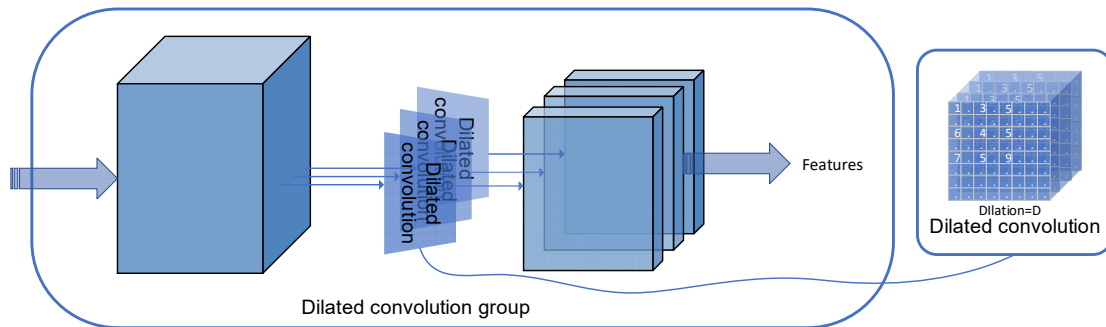


Figure 3. Dilated convolution group (Dconv group), where D denotes the dilation factor.

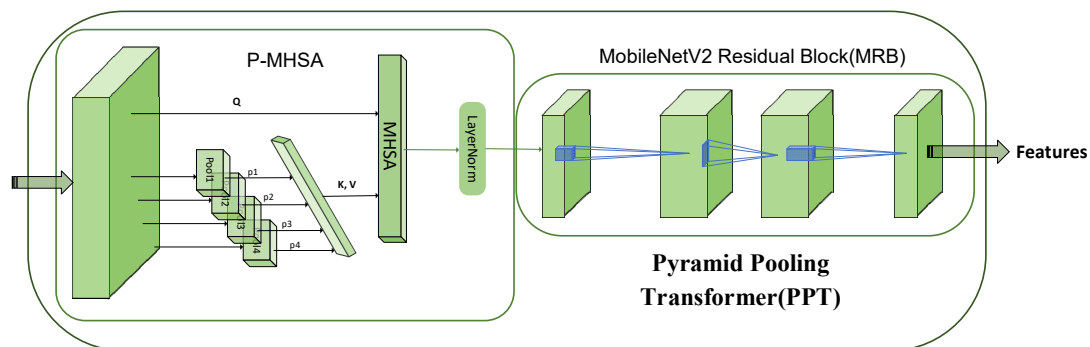


Figure 4. Pyramid Pooling Transformer (PPT). The PPT in this study, in which the P-MHSA and the MRB are not skip-connected, differs from the traditional PPT. In the MobileNetV2 Residual Block (MRB) subgraph, the thickness of the shaded convolution blocks in the feature block indicates the relative number of channels. The MRB computes the feature block using the high channel convolution for upscaling, then each channel of the feature block is calculated for the low channel convolution alone and finally for the high channel convolution for downscaling.

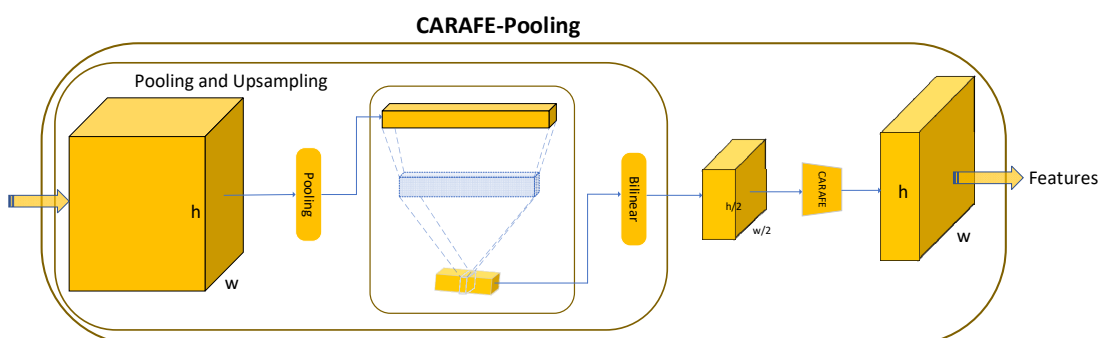


Figure 5. CARAFE pooling structure diagram.

The 1×1 convolutional layer is represented by Equation (1). For brevity, the above equation does not declare normalization, activation functions, or other operations, so they are added here. Equation (1) is followed by BatchNorm batch normalization and the Relu activation function.

$conv_{1 \times 1}$ has been retained to make MBA-ASPP more accurate with ResNet [32] short-circuiting.

$$P_i = conv_{rate=d_i}(X), i = 2, 3, 4 \quad (2)$$

where d_i for $rate = d_i$ ($i = 2, 3, 4$) represents the dilation factor. The atrous convolutional group is represented by Equation (2). For brevity, the above equation does not declare normalization, activation functions, or other operations, so they are added here. Equation (2) is followed by BatchNorm batch normalization and the Relu activation function.

The atrous pyramid is retained, in which MBA-ASPP can extract dilated features.

$$P_j = CARAFE - pooling_{type=t_j}(X), j = 5, 6, 7 \quad (3)$$

where t_j for $type = t_j$ ($j = 5, 6, 7$) represents the pooling method (global average pooling [18], global maximum pooling, and deep convolution [33]). CARAFE pooling is represented by Equation (3). Here, Equation (3) consists of global pooling (avg, max, and conv), a CNN 1×1 adjustment for dimensionality reduction, BatchNorm for batch normalization, Relu for the activation function, bilinear interpolation [34], and CARAFE [23].

ASPP's pooling layer is formed by bilinear interpolation to upsample operation and bring the feature map back to its original resolution, but the bilinear interpolation [34] guides the upsampling by the sub-spatial distance of the pixels and does not have learnability. Therefore, this study uses the CARAFE operator [23] to assist bilinear interpolation to complete the upsampling operation. Specifically, on top of the pooling layer, the feature map is restored to half the original resolution using bilinear interpolation, and then the feature map is restored to its original resolution using the CARAFE operator to make it learnable, this is called CARAFE pooling. One of the CARAFE operators [23] is a lightweight upsampling operation, which has a lower computational complexity compared to upsampling operations such as STN [35] and DCN [36].

$$P_k = MRB(P - MHSA_{type=t_k}(X)), k = 8, 9, 10 \quad (4)$$

where t_k in $type = t_k$ ($k = 8, 9, 10$) represents the pooling method (global average pooling, global maximum pooling, and deep convolution). The Pyramid Pooling Transformer (PPT) is represented by Equation (4). Unlike conventional PPT, the PPT used in this study consists of P-MHSA and MBR, but neither P-MHSA nor MBR was skip-connected. The MRB [24] uses the Hardswish [37] activation function with the connection order: CNN 1×1 , Hardswish, CNN 3×3 , Hardswish, CNN 1×1 . The dimensional transformation is omitted.

For the PPT, this study uses the PPT of P2T (Pyramidal Pooling Transformer network) [28], but it is not skip-connected. Context information is extracted and forward propagation of MRB is used by the PPT. The PPT then cascades to other branches so that other branches can learn the contextual information from the PPT. The PPT [28] used in this study does not learn because no skip connections are used. Its task is to extract the contextual information that helps other branches learn the contextual information. However, MRB has learned that IBR can learn the features of other branches. Introducing the PPT, the MBA-ASPP has a global attention mechanism and is richer in contextual information than the ASPP. MASPP can also be used for large image segmentation. For example, mimicking the Swin Transformer [21] and P2T [28], the $C \times 256 \times 256$ feature map is divided into four $C \times 128 \times 128$ sub-feature maps. First, the sub-feature maps are converted into $C \times 64 \times 64$ feature maps, then the sub-feature maps are passed to the branch of MASPP for processing and finally, the processed sub-feature maps from MASPP are spliced into $C \times 128 \times 128$ feature maps. After several layers of MASPP, a small feature map will be obtained. This is useful for the model to extract the most critical local features that can determine the category. However, there is no research on semantic segmentation of large images in this study, only speculation on large image segmentation.

4.2. Cascaded Adaptive Parameters

Existing multi-branch feature extraction networks, such as AFP [20], GooLeNet [38], and OSN (Omni Scale Net) [30,31], introduce different branches to extract diverse features to enrich the feature map. AFP uses weighted summation with adaptive weights to fuse features but sacrifices feature dimensionality to obtain a rich feature map. GooLeNet achieves this by cascading with a fixed dimension ratio and OSN uses full-scale features and a through-unity aggregation gate. MBA-ASPP maintains the dimensionality of the features and facilitates or suppresses the weight ratio of different branches by adaptive parameters, so MBA-ASPP preserves the dimensionality of the cascaded features and performs weighted summation.

First, the number of branches of MASPP is ten, and more features with different sensory fields can be extracted to enrich the feature map in the case of multiple branches. Equation (1) preserves the original features to the maximum extent in the same manner as ResNet [32], and the dilated convolutional group denoted by Equation (2) gives MBA-ASPP the ability to extract dilated information. CARAFE pooling, denoted by Equation (3), can extract pooling information with learning. PPTs denoted by Equation (4) can provide a global attention mechanism. If the number of branches approaches 10, this may lead to the same proportion of less important branches to important branches, and a long training period is required to suppress this effect.

To solve the above problems, this study proposes cascaded adaptive parameters (CAPs). In the case of cascading, the weight ratios of different branches are learned by introducing adaptive parameters so that the model can increase the weight ratios of specific branches by itself and decrease the weight ratios of specific branches by itself according to the feature distribution of the data. The fusion equation is as follows:

$$P = \text{Concat}\{P_1 \times a_1, P_2 \times a_2, \dots, P_i \times a_i, \dots, P_{10} \times a_{10}\} \quad (5)$$

where $a_i = \frac{e^{w_i}}{\sum_{i=1}^n e^{w_i}} \times n$ ($i = 1, 2, 3, \dots, n$) is the CAP, w_i is the weight and has an initial value of 1, and n is the number of branches.

After extensive experiments, it was found that the error of the experimental data obtained by using CAPs trained under the same conditions was due to chance uncertainty. To overcome this chance uncertainty, this experiment uses PCS, which can effectively suppress the effect of chance uncertainty.

$$\phi = f(x) \quad (6)$$

where x is the independent variable and $f(x)$ is $f(x) = \infty$, $f(x) = 1$, $f(x) = 2^x$ ($x = 0, 1, 2, \dots$). Equation (6) represents the interval $f(x)$ in training cycles to initialize w_i . For example, when training for 46 cycles (46 epochs equals 30 k iterations), $f(x) = \infty$ means no initialization of w_i during training; $f(x) = 1$ means initialization of w_i for each training cycle (46 times in total); and $f(x) = 2^x$ denotes initialisation of w_i , where $\lfloor \log_2 46 \rfloor = 5$ times. w_i is initialized when epoch = (2, 4, 8, 16, 32); the initialization intervals are increasingly large and the growth rate is exponential. When the number of cascaded adaptive parameters is ten, the image is too large and dense to accurately show them. For the sake of readability, the periodic coefficient strategy is simulated using three CPAs. The change in CPAs using the periodic coefficient strategy is shown in Figure 6.

Inspired by Cyclical Learning Rates (CLRs) [39], during network training, the model inevitably falls into local minima or saddle points several times, and the use of these CLRs (learning rates varying between maximum and minimum values) helps the model to move out of local minima or saddle points faster. This study uses Equation (6), which periodically adjusts the weight ratio of each branch (the weight ratio is initialized to 1 for each branch), to help the model move out of local minima or saddle points due to chance uncertainty. However, adjusting the weight ratio of each branch may cause the model to jump from one critical point to another (local minima or saddle points). Therefore, the main idea of this study is to periodically adjust the weight ratio of each branch using Equation (6). This is

done to maintain a large learning rate at all times, so that if the model is at a local minimum or saddle point for a long time due to chance uncertainty but is unable to move away (or if the model is not at a critical point), the model will move away from that critical point (or out of that position) when the periodic function takes effect. The model may experience three situations: 1, not immediately falling into the next critical point; 2, falling into the next critical point, relying on a larger learning rate to disengage; and 3, falling into the next critical point, not relying on a larger learning rate to disengage and waiting for periodic adjustment of each branch weight ratio. To maintain a large learning rate at all times, this study uses the learning rate strategy polylr. The learning rate (lr) is set to 60 k by the formula $lr' = lr \times \left(1 - \frac{iter}{max_iter}\right)^{power}$, while the maximum number of iterations, $iter$, is only 30 k and the power is set to 0.9 so that the decay rate of the learning rate (lr) is small and the minimum value is half of the initially set learning rate. The optimizer in this study uses Stochastic Gradient Descent (SGD), a method for finding optimal parameter configurations for machine learning algorithms. As shown in Figure 7, the graphs of $f(x) = 2^x$ have the fastest convergence in terms of the loss (compared to the CLR, which acts directly on the learning rate, which is not significant), and the periodic coefficient strategy (PCS) has a domain of only n parameters.

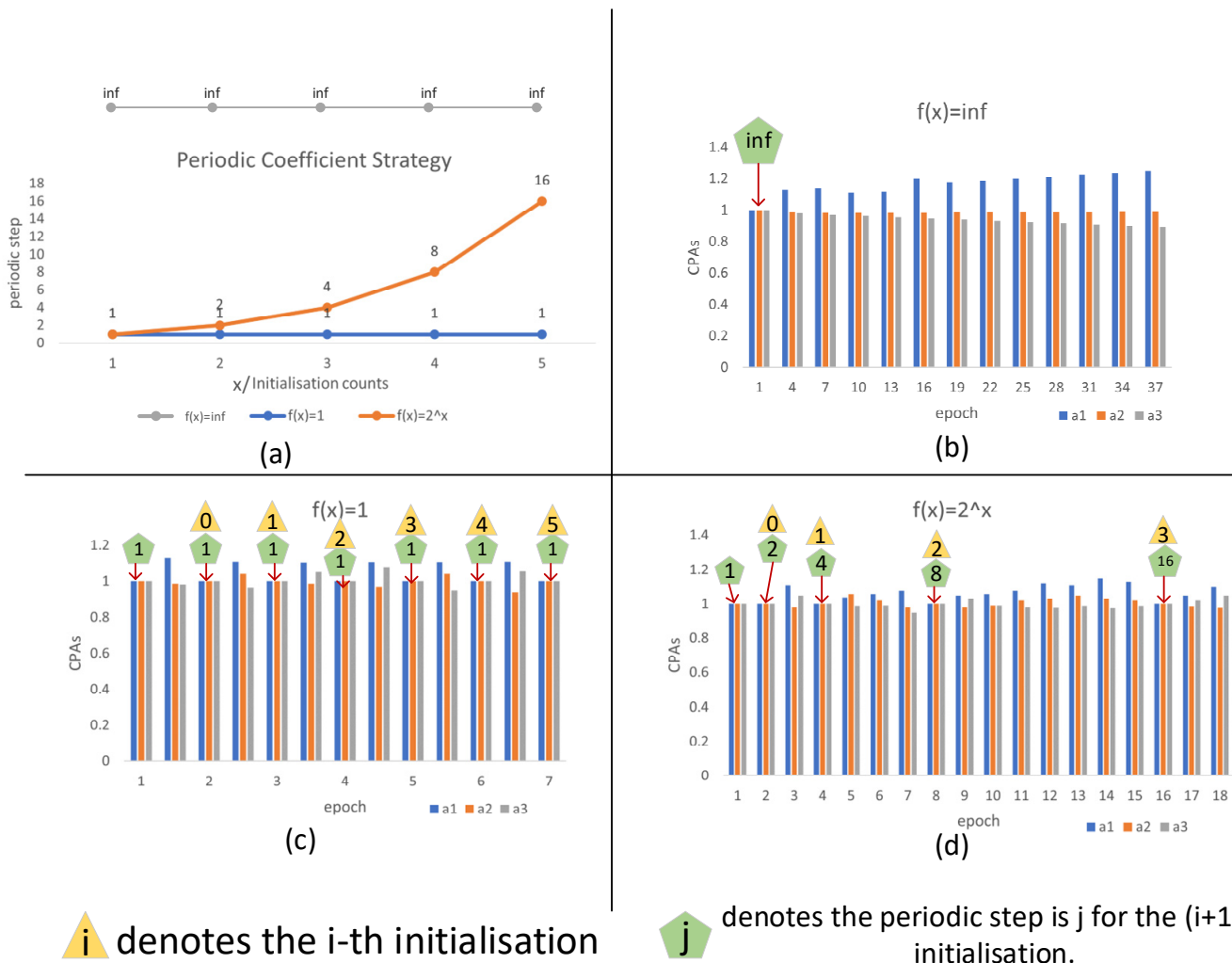


Figure 6. (a) shows the change in the periodic step of the diagram of CPAs using the periodic coefficient strategy; (b–d) show the variation plots of the CPAs using the periodic coefficient strategy of $f(x) = \infty$, $f(x) = 1$, and $f(x) = 2^x$, respectively. inf denotes ∞ .

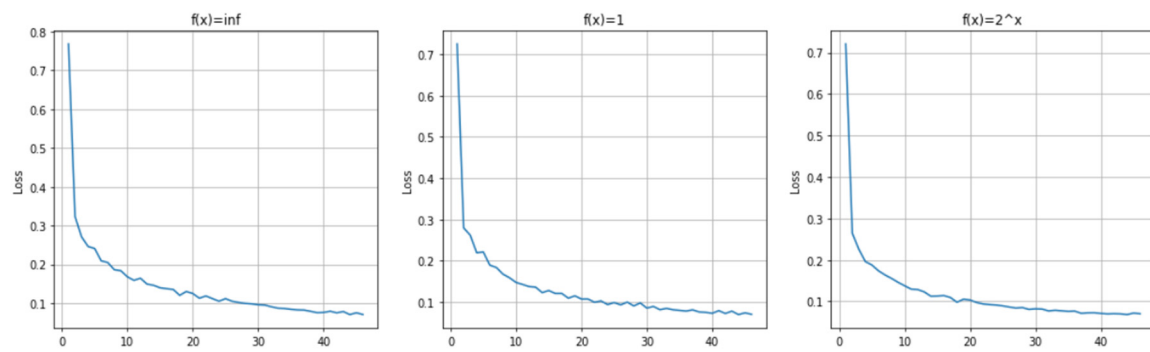


Figure 7. The plot of the effect of the strategy of the validated period coefficient on the loss.

4.3. Overlap MobileNetV2 Residual Block

Considering that MBA-ASPP uses CAPs to induce the model to automatically suppress or promote the weight ratio of each branch without changing its feature dimension, the output of the model must be compressed in the feature dimension. In the field of image segmentation, a convolutional kernel of 1×1 is generally used with a convolutional compression dimension, resulting in a certain loss of features. The use of linear transformations offsets the shortcomings of a convolutional kernel of 1×1 , but SPPNet [27] uses multiple pooling layers stitched together to obtain a very long sequence vector, which will undoubtedly increase the difficulty of model training. Therefore, this study introduces MRB [24,28], and thus proposes Overlap MobileNetV2 Residual Blocks (OMRBs). Overlap convolution and MRB [24] are employed to construct an approach of vertical and horizontal modeling. As shown in Figure 8, this process is as follows:

$$(x_h, x_w) = \text{Overlap convolution}(P) \quad (7)$$

$$x_w = \text{MRB}(x_w) \quad (8)$$

$$x_{out} = x_h + x_w \quad (9)$$

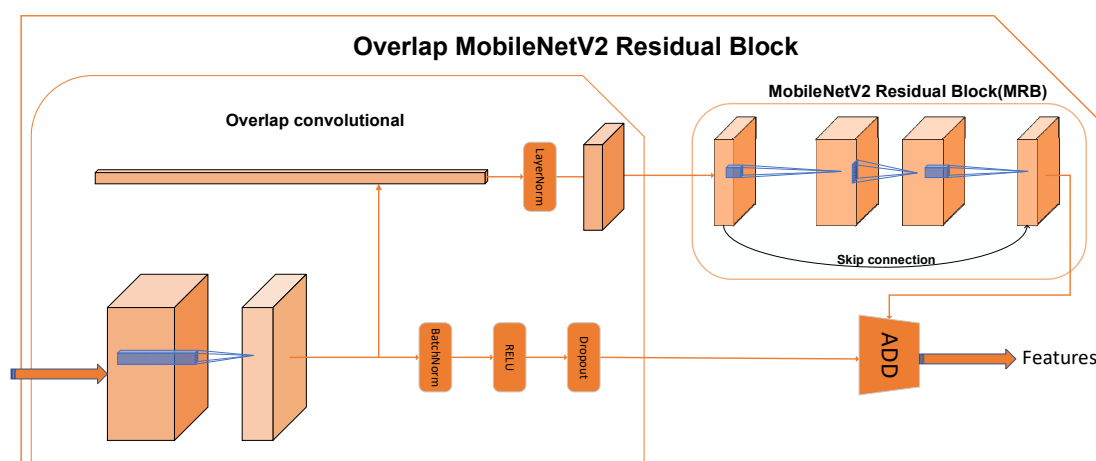


Figure 8. OMRB structure diagram. In the MobileNetV2 Residual Block (MRB) subgraph, the thickness of the shaded convolution blocks in the feature block indicates the relative number of channels. The MRB computes the feature block using the high channel convolution for upscaling, then each channel of the feature block is calculated for the low channel convolution alone and finally for the high channel convolution for downscaling.

The dimensional transformation operation has been omitted for brevity.

To explain global longitudinal and lateral modeling, Equations (7) and (8) are interpreted in terms of convolution operations. Let the input layer channels, output layer channels, and hidden layer channels be c_{in} , c_{out} , and c_{hid} respectively. To define this simply, the channel affected by the channel is considered as vertical and the one affected by the width and height is considered as horizontal.

Equations (7) can be simplified to a single convolution operation. It may be useful to set the input as P , the convolution as A , and the output as X_h , whose shapes are $[c_{in}, w, h]$, $[c_{out}, c_{in}, 1, 1]$, and $[c_{out}, w, h]$. P can be represented by $[p_{k,i,j}]$ ($k = 1, 2, \dots, c_{in}$, $i = 1, 2, \dots, w$, $j = 1, 2, \dots, h$). A is represented by the c_{out} sub-convolution A_l ($l = 1, 2, \dots, c_{out}$), the shape of the sub-convolution A_l is $[c_{in}, 1, 1]$, and the sub-convolution A_l can be represented by $[a_k]$ ($k = 1, 2, \dots, c_{in}$). The sub-convolution A_l performs the convolution operation on P , obtaining x_{h_l} of shape $[w, h]$, where x_{h_l} can be represented by $[x_{h_l,i,j}]$. The formula for its operation is as follows:

$$x_{h_l,i,j} = \sum_{k=1}^{c_{in}} p_{k,i,j} \times a_k \quad (10)$$

The sub-convolution A_l of P uses Equation (10) to obtain x_{h_l} , and the sub-convolution A of P uses Equation (10) c_{out} times to obtain c_{out} x_{h_l} , where X_h can be denoted by $[x_{h_l}]$. It can be observed that $x_{h_l,i,j}$ consists of the longitudinal matrix of P which performs the convolution operation with the sub-convolution A_l , i.e., $x_{h_l,i,j}$ is related to $p_{k,i,j}$ (i, j fixed, $k = 1, 2, \dots, c_{in}$) in the longitudinal direction. x_{h_l} represents one longitudinal modelling of the input P and X_h represents the c_{out} sublongitudinal modelling of P . X_h is then copied and converted to a vector to give $X_{w'}$. Normalization of $X_{w'}$ is LayerNorm. The normalization and activation functions of X_h are BatchNorm and Relu, respectively. Finally, the dropout layer is used for X_h to reduce the overfitting of the neural network.

Equation (8) can be simplified into three convolution operations, the first and third convolution operations are ascending and descending operations, respectively, and the second is a convolution operation for transverse modeling. The first convolution operation can be represented as converting $X_{w'}$ with shape $[c_{out}, w, h]$ to X'_w with shape $[c_{hid}, w, h]$, and the third convolution operation can be represented as converting the shape of X_w from $[c_{hid}, w, h]$ to $[c_{out}, w, h]$. Next, the second convolution operation—transverse modeling—is explained. It may be useful to define convolution B , whose shape is $[c_{hid}, 1, n, m]$. B consists of c_{hid} sub-convolutions B_q , and B can be represented as $[B_q]$ ($q = 1, 2, \dots, c_{hid}$). The shape of the subconvolution B_q is $[n, m]$ (both n and m are odd integers), and the sub-convolution B_q can be represented as $[b_{s,t}]$ ($s = 1, 2, \dots, n$, $t = 1, 2, \dots, m$). X'_w can be expressed as $[x'_{q,i,j}]$. In the case of a fixed channel q , the sub-convolution B_q performs the convolution operation with the transverse matrix corresponding to X'_w to obtain x_{w_q} of shape $[w, h]$, where x_{w_q} can be expressed as $[x_{w_q,i,j}]$. The arithmetic formula is as follows:

$$x_{w_q,i,j} = \sum_{s=1}^n \sum_{t=1}^m x'_{q,i-(\lceil n/2 \rceil - s), j-(\lceil m/2 \rceil - t)} \times b_{s,t} \quad (11)$$

where $\lceil \frac{n}{2} \rceil$ and $\lceil \frac{m}{2} \rceil$ represent rounding up by half for n and m , respectively.

In the corresponding channel q , the sub-convolution B_q to $x'_{q,i,j}$ using Equation (11) yields x_{w_q} , and the sub-convolution B_q of B to the corresponding channel q of X'_w using Equation (11) yields c_{hid} individual x_{w_q} , where X_w can be represented by $[x_{w_q}]$. It can be observed that $x_{w_q,i,j}$ consists of the matrix of the corresponding channel of X'_w which performs the convolution operation with the sub-convolution B_q of the corresponding channel, independent of the matrices of the other channels, i.e., $x_{w_q,i,j}$ is related to $x'_{q,i,j}$ only in the transversal direction. x_{w_q} represents one transversal modelling of X'_w and X_w represents the c_{hid} sub-horizontal modelling of X'_w .

4.4. Realisation Details

The settings for the different branches of MASPP are as follows. Based on the ASPP, the $conv_{1 \times 1}$ and $conv_{rate=k}$ branches have an output dimension of 256 and a $conv_{rate=k}$ expansion factor of {2,4,8}. The CARAFE pooling branch, which has an output dimension of 256, is first restored using bilinear interpolation [34] to half of the resolution and then restored to the original resolution using the CARAFE operator. The PPT branch sets the pooling ratio to {1,2,4,8}, but in this study, the input is first dimensionally compressed to 1/4 of the original dimension using a 1×1 convolution and then placed in the PPT, whose output dimension is also set to 256. Finally, the input layer is the output of ResNet [32], which has a dimension of 2048, and the output dimension of MASPP is 2560.

The cascaded adaptive parameters are defined as follows. After extensive experiments, it was found that the error of the experimental data obtained by using CAPs trained under the same conditions was affected by chance uncertainty. A periodic coefficient strategy (PCS) was used before each epoch.

The OMRB setup is as follows. MRB uses the Hardswish [37] activation function rather than GELU [40].

The initialization was as follows. The weights of Conv2d and Linear are initialized using the normal distribution Kaiming Normal with a mean of 0 and a variance of 1. The weights of BatchNorm2d and LayerNorm are assigned a value of 1 and the bias is assigned a value of 0.

5. Experiment

5.1. Experimental Environment and Dataset

The experiments were performed under Linux using the Pytorch 1.9.1 deep learning framework and CUDA version 11.1, with a Tesla V100 GPU.

Experiments were conducted using an image segmentation dataset as the experimental data for semantic segmentation, the PASCAL VOC 2012 dataset [41], and a complementary experiment using an image classification dataset (the CIFAR-100 dataset) [42].

The PASCAL VOC 2012 [41] dataset, with 20 categories in the segmentation task, each filled with a specific color, was used for semantic segmentation with a training data sample of 10,582 images and a test data sample of 1449 images.

The CIFAR-100 [42] dataset has 100 categories in the classification task, with each category containing 500 training images and 100 test images.

5.2. Experimental Details

An ImageNet [43] pre-trained model was used in the experiments with ResNet50 [32], with output_stride = 16 and batchsize = 16. The optimizer used SGD and set the learning rate of MAB-ASPP, lr, as 0.01. The learning rate of ResNet50 was 0.1 times the learning rate of MAB-ASPP's lr, and the learning rate strategy used poly lr, with max_iters = 60 k and power = 0.9, and training was iterated 30 k (epoch = 46) times. Any changes in hyperparameters or other settings will be specified.

5.3. Experimental Results

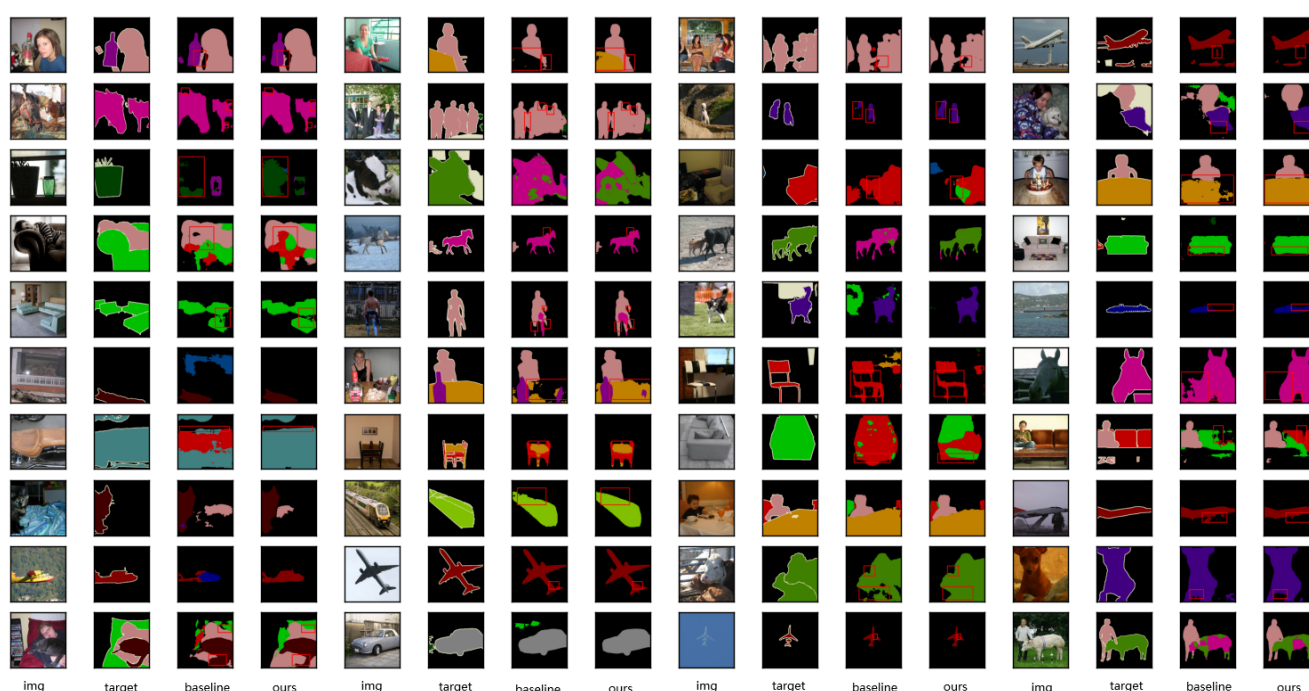
For ease of description, Deeplabv3plus-MBA represents the replacement of ASPP with MBA-ASPP based on Deeplabv3plus. To evaluate the performance of MBA-ASPP, the method proposed in this study was compared with other superior models. As shown in Table 1, Deeplabv3plus-MBA was improved by 1.29% over Deeplabv3plus on the PASCAL VOC 2012 dataset. None of the experiments were performed with MS, Flip, and COCO.

Table 1. Performance on the PASCAL VOC 2012 test set.

Method	MIoU
DeepLabv2-CRF [9]	77.69
PSPNet [19]	77.13
Deeplabv3 [10]	77.21
Deeplabv3plus [10]	77.81
Deeplabv3plus-MBA	79.10

5.4. Analysis of Results

As shown in Figure 9, Deeplabv3plus-MBA exhibited a significant performance on the PASCAL VOC 2012 dataset, using Deeplabv3plus [10] as the baseline. The PASCAL VOC 2012 dataset is a multi-category dataset and has significant improvements over the PASCAL VOC 2012 dataset, validating the side accuracy of this study.

**Figure 9.** Comparison of the effects of Deeplabv3plus-MBA with those of the baseline. Ours represents the prediction plot for Deeplabv3plus-MBA.

5.4.1. Parameter Size

Most models get better as the number of layers increases and the parameter size increases. Deeplabv3plus-MBA does not get better by simply adding more branches and increasing the parameter size, which demonstrates the superiority of MBA-ASPP, as shown in Table 2. The task of this study was to improve Deeplabv3plus's ASPP, so Deeplabv3plus-wide was designed to obtain a parameter size that matched Deeplabv3plus-MBA by setting the output dimension of each branch of the ASPP to twice its original size (originally 256, doubled to 512). Deeplabv3plus was replaced by baseline. In comparison to baseline, Deeplabv3plus-MBA improves the Mean Pixel Accuracy (MPA), the Frequency-Weighted Intersection over Union (FWIoU), and the Mean Intersection over Union (MIoU) by 1.88%, 0.66%, and 1.48%, respectively. The Mean Pixel Accuracy represents the proportion of correctly categorized pixels for each class separately and their cumulative average. The Mean Intersection over Union is a commonly used evaluation metric to measure the similarity between predicted results and real labels. The Frequency-Weighted Intersection over Union is an improvement to the MioU (Mean Intersection over Union), a method that sets weights for each class based on how often it appears.

Table 2. Parameter scale comparison.

Method	Total Params	MPA	FWIoU	MIoU
Deeplabv3plus (baseline)	39,761,845	87.24	89.68	77.62
Deeplabv3plus-wide	55,296,437	87.36	89.63	77.73
Deeplabv3plus-MBA	56,071,599	89.12	90.34	79.10

5.4.2. Ablation Experiments

The effect of each module is first shown, and then ablation experiments were performed for each module in turn, as shown in Table 3. Using Deeplabv3plus [10] as the baseline, it can be observed that the effects of MASPP, CAPs, and OMRB on the baseline are all positive; each module exhibits a small improvement over the baseline, while the improvement of all modules combined is particularly significant.

Table 3. Ablation experiments on Deeplabv3plus-MBA.

Serial	ASPP	MASPP	CAPs	OMRB	MPA	FWIoU	Miou
1	T				87.24	89.68	77.62
2		T			87.83	89.86	78.09
3	T		T		87.21	89.85	78.05
4	T			T	87.74	89.93	77.83
5		T	T	T	88.41	90.42	79.10

1. Multibranch Atrous Spatial Pyramid Pooling

To verify the accuracy of MASPP (multibranch atrous spatial pyramidal pooling), we performed ablation and combination experiments on CARAFE pooling and the PPT (Pyramid Pooling Transformer) while retaining the other modules. The PPT with type Conv was found to exhibit a high accuracy improvement in local positive feedback compared to other CARAFE pooling branches and PPT branches, as shown in Table 4; exhibit a high accuracy improvement in global negative feedback compared to eliminating other CARAFE pooling branches or PPT branches alone. Removing alone a Conv-type PPT branch lead to the highest accuracy, as shown in Table 5. We hypothesize that the addition of a global attention mechanism when adding a Conv-type PPT alone, shown for example in Table 4, compensates for the lack of global contextual information in ASPP. In Table 5, for example, after elimination of the Conv-type PPT alone, the CARAFE pooling (Avg, Max, and Conv) branch group provides global pooling information with learning and the Avg-type PPT (or Max) provides the global contextual mean. The Avg-type PPT (or Max) provides the global contextual mean (or maximum information), highlighting edge contours and masks of interest, while the Conv-type PPT learns global contextual information from the large convolutional kernel, which only serves to complement the global mean and maximum, and has no significant effect on capturing edge contours and highlighting masks of interest, and therefore does not improve the method much.

Table 4. The contribution of each branch of MASPP is verified separately based on Deeplabv3plus-MBA.

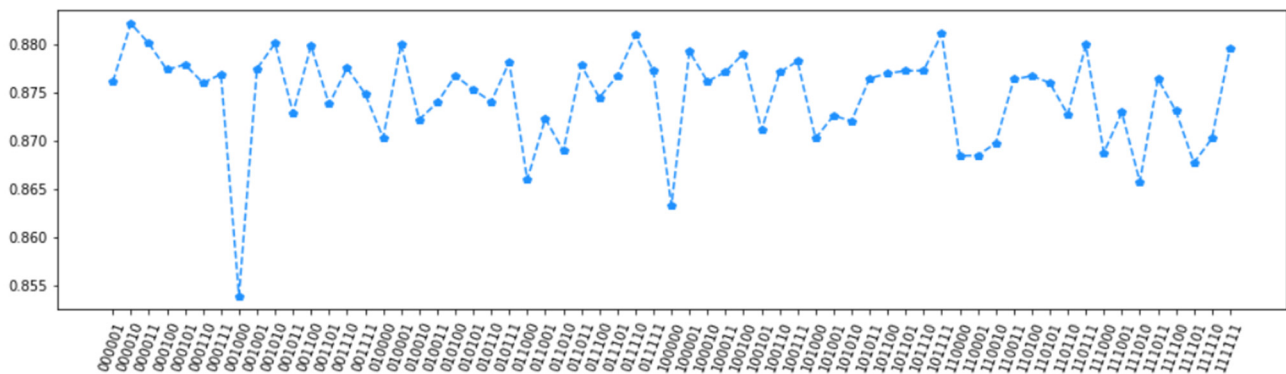
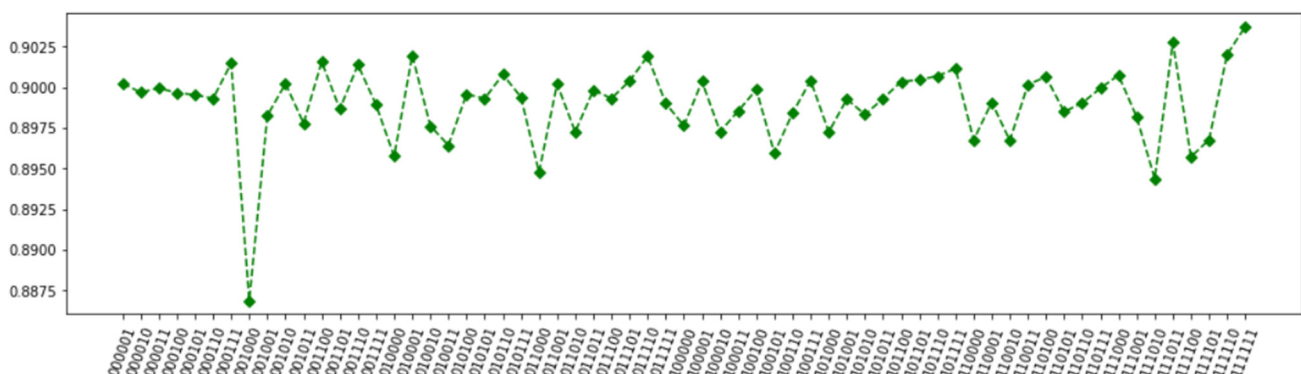
Serial	CARAFE Pooling			PPT			MPA	FWIoU	MIoU
	avg	max	conv	avg	max	conv			
1	T						86.32	89.76	77.61
2		T					87.03	89.57	77.45
3			T				85.39	88.68	75.23
4				T			87.73	89.96	78.48
5					T		88.21	89.96	78.42
6						T	87.67	90.03	78.44

Table 5. Validation of partial branch combinations for MASPP based on Deeplabv3plus-MBA.

Serial	CARAFE Pooling			PPT			MPA	FWIoU	MIoU
	avg	max	conv	avg	max	conv			
1		T	T	T	T	T	87.72	89.90	78.18
2	T		T	T	T	T	88.10	90.11	78.38
3	T	T		T	T	T	87.99	89.99	78.29
4	T	T	T		T	T	87.64	90.27	78.72
5	T	T	T	T		T	86.77	89.67	77.54
6	T	T	T	T	T		87.02	90.20	78.91

To allow for better detection of local negative feedback and global positive feedback, the following table is given.

After extensive experimentation, we found that the best results were obtained using both CARAFE pooling and the PPT. As shown in Figures 10–12, all combinations are represented in a 6-bit binary code, with 0 for False and 1 for True. Their order, from high to low, can be represented as CARAFE poolings (Avg, Max, and Conv) and PPTs (Avg, Max, and Conv).

**Figure 10.** Verification of MPA for all branch combinations of MASPP.**Figure 11.** Verification of FWIoU for all branch combinations of MASPP.

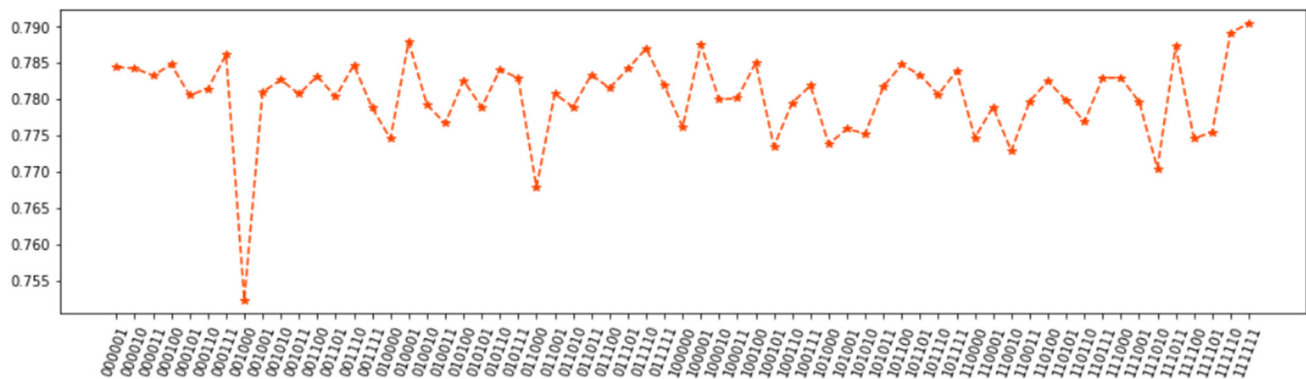


Figure 12. Verification of MIoU for all branch combinations of MASPP.

2. Cascaded Adaptive Parameters

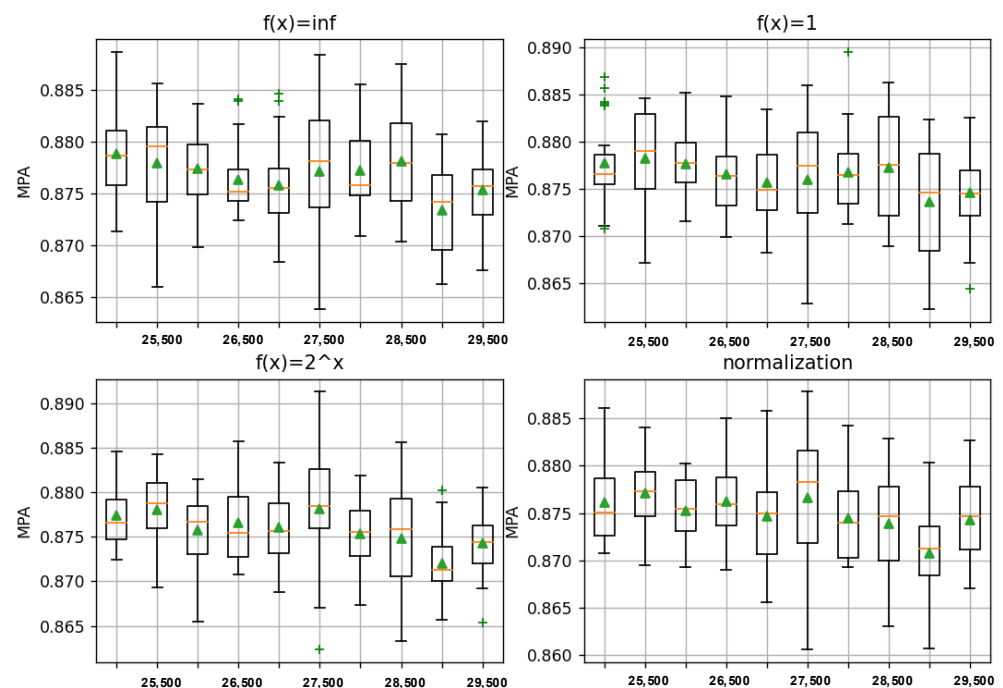
Branches are added in MBA-ASPP by different convolutional layers, CARAFE pooling, and the PPT (Pyramidal Pooling Transformer), which results in features with different sensory fields and contextual information to enrich the feature map. Based on cascaded adaptive parameters (CAPs), the model automatically suppresses or promotes different branches to assign the weight ratio of different branches.

The validation of the CAP module under the condition of using a period coefficient strategy is demonstrate in the following.

- (1). There is an improvement. The experimental results show that it can be observed that for MBA-ASPP using CAPs and normalized weights, respectively, a boost in CAPs over normalized weights exists but is not significant, as shown in Table 6. With the iterations set to 30 k (epoch = 46) and the period coefficient method $f(x) = 2^x$ and CPAs, an MPA of 89.12%, an FWIoU of 90.34%, and an MioU of 79.10% were produced, with MPA, FWIoU, and MioU improving by 0.43%, 0.05%, and 0.18%, respectively. Setting the iterations to 60 k (epoch = 91) and using the period coefficient strategy $f(x) = 2^x$ and CPAs, an MPA of 88.57%, an FWIoU of 90.56%, and an MioU of 79.51% were obtained, where MPA, FWIoU, and MioU were improved by 0.50%, 0.12%, and 0.35% respectively.
- (2). It is more stable. Training a neural network by randomly sampling the data will cause data fluctuations. This is due to SGD, which can lead the model into a local minimum or saddle point, so data fluctuations often exist in the neural network training phase. As shown in Figure 13, the MPA (Mean Pixel Accuracy) box plots of data fluctuations found with $f(x) = 1$ have the smallest range of data fluctuations and the best means. In Figure 14, the FWIoU (Frequency-Weighted Intersection over Union) box plots of data fluctuations found with $f(x) = 2^x$ have the best means and the fewest outliers. In Figure 15, the MIoU (Intersection over Union) data fluctuation plot found with $f(x) = 1$ have the least number of blank spaces in the Savitzky–Golay fit line (adjacent [max, mean] or [min, mean] are not adjacent to each other, the Savitzky–Golay fits to large blank areas between the maximum and minimum). However, it can be concluded from the experimental data in Table 6 that the periodicity coefficient strategy $f(x) = 2^x$ outperforms the others. Using the average value as the evaluation criterion, the periodicity coefficient strategy ($f(x) = 2^x$)’s MPA, FWIoU, and MioU are 88.481%, 90.333%, and 79.071%, respectively, while the periodicity coefficient strategy ($f(x) = 1$)’s MPA, FWIoU, and MioU are 87.958%, 90.240%, and 78.785%, respectively. The periodicity coefficient strategy ($f(x) = 2^x$) has a greater MPA, FWIoU, and MioU than the periodicity coefficient strategy ($f(x) = 1$).

Table 6. Validation of CPAs and periodicity coefficient strategies and normalized weights. * represents iters = 60 k.

Method	$conv_{1 \times 1}$	$conv_{rate=k}$			CARAFE-Pooling			PPT			MPA	FWIoU	MIoU
		2	4	8	avg	max	conv	avg	max	conv			
$f(x) = \infty$	0.8853	1.3303	1.1153	1.1166	0.9943	0.9977	0.9841	0.8504	0.8584	0.8670	88.41	90.42	79.10
$f(x) = \infty$	0.8122	1.3325	1.1469	1.1425	0.9972	1.0009	0.9818	0.8413	0.9202	0.8240	87.92	90.35	79.05
$f(x) = \infty$	0.8116	1.2841	1.1491	1.1509	0.9989	1.0053	0.9823	0.8222	0.9621	0.8329	88.18	90.18	78.81
$f(x) = \infty$	0.8175	1.2785	1.1401	1.1529	1.0003	1.0051	0.9813	0.8582	0.9448	0.8209	88.83	90.21	78.79
$f(x) = \infty$	0.8028	1.2855	1.1435	1.1361	1.0016	1.0070	0.9826	0.8275	0.9843	0.8286	87.48	90.11	78.53
* $f(x) = \infty$	0.7624	1.2680	1.1393	1.1165	1.0002	1.0060	0.9784	0.8853	1.0027	0.8407	87.51	90.47	79.12
$f(x) = 1$	0.9998	1.0006	1.0010	0.9997	0.9999	0.9999	0.9999	1.0003	0.9987	0.9996	87.96	90.37	79.04
$f(x) = 1$	0.9908	1.0028	1.0029	1.0039	1.0000	0.9999	0.9999	1.0000	1.0028	0.9965	88.47	90.23	78.89
$f(x) = 1$	0.9929	1.0117	1.0035	1.0007	0.9998	1.0001	0.9996	0.9952	1.0071	0.9889	87.55	90.19	78.81
$f(x) = 1$	0.9994	0.9996	1.0008	1.0004	0.9999	1.0000	0.9999	1.0001	0.9997	0.9996	88.07	90.17	78.61
$f(x) = 1$	0.9822	1.0087	1.0105	0.9988	0.9999	0.9999	0.9998	0.9952	1.0040	1.0006	87.90	90.02	78.32
* $f(x) = 1$	0.9983	1.0016	1.0011	0.9988	1.0000	0.9999	1.0000	0.9990	1.0005	1.0005	87.80	90.46	79.03
$f(x) = 2^x$	0.8903	1.1339	1.0563	1.011	0.9984	0.9979	0.9963	0.9702	0.9714	0.9732	89.12	90.34	79.10
$f(x) = 2^x$	0.8758	1.1374	1.0499	1.0028	0.9991	0.9988	0.9960	1.0095	0.9610	0.9693	87.73	90.38	79.09
$f(x) = 2^x$	0.8801	1.1431	1.0484	1.0136	0.9990	0.9983	0.9956	0.9687	0.9767	0.9761	88.56	90.35	79.05
$f(x) = 2^x$	0.8764	1.1204	1.0551	1.0037	0.9998	0.9984	0.9963	0.9979	0.9794	0.9720	88.40	90.20	78.85
$f(x) = 2^x$	0.8757	1.1311	1.0561	1.0081	0.9986	0.9972	0.9967	0.9803	0.9792	0.9765	88.51	90.17	78.83
* $f(x) = 2^x$	0.8732	1.0851	1.0406	0.9815	0.9984	0.9984	0.9970	1.0248	0.9968	1.0037	88.57	90.56	79.51
normalization	1	1	1	1	1	1	1	1	1	1	88.69	90.29	78.92
normalization	1	1	1	1	1	1	1	1	1	1	88.77	90.17	78.90
normalization	1	1	1	1	1	1	1	1	1	1	88.40	90.13	78.78
normalization	1	1	1	1	1	1	1	1	1	1	87.77	90.25	78.74
normalization	1	1	1	1	1	1	1	1	1	1	87.96	90.11	78.49
* normalization	1	1	1	1	1	1	1	1	1	1	88.07	90.44	79.16

**Figure 13.** Validation of MPA for CAPs and PCS and MPA for normalized weights, where inf denotes ∞ . Green triangles representing means and green '+' symbols representing outliers.

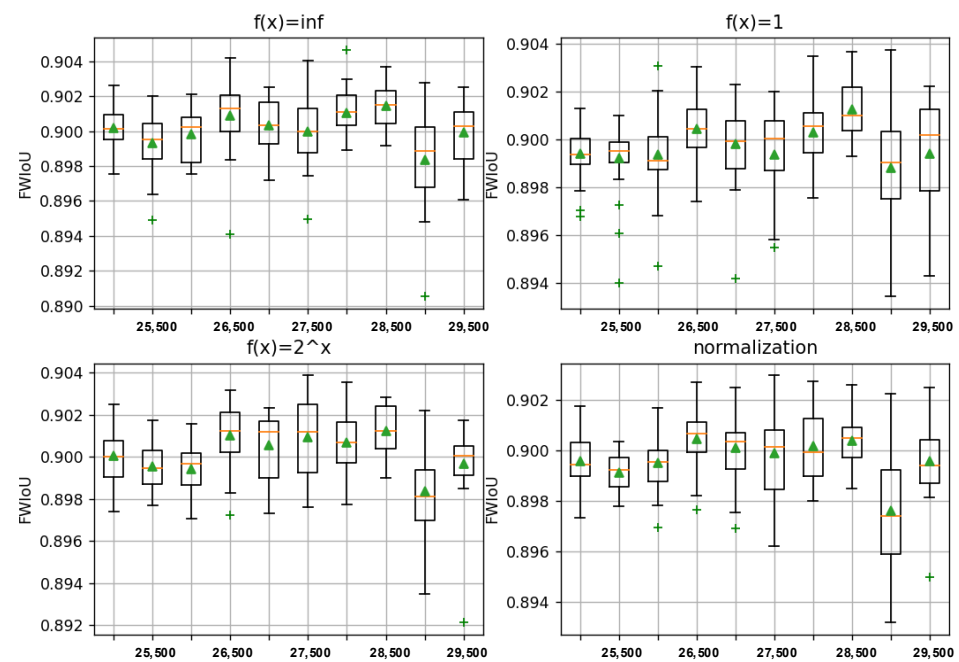


Figure 14. Validation of FWIoU for CAPs and PCS and FWIoU for normalized weights, where $\inf$ denotes ∞ . Green triangles representing means and green '+' symbols representing outliers.

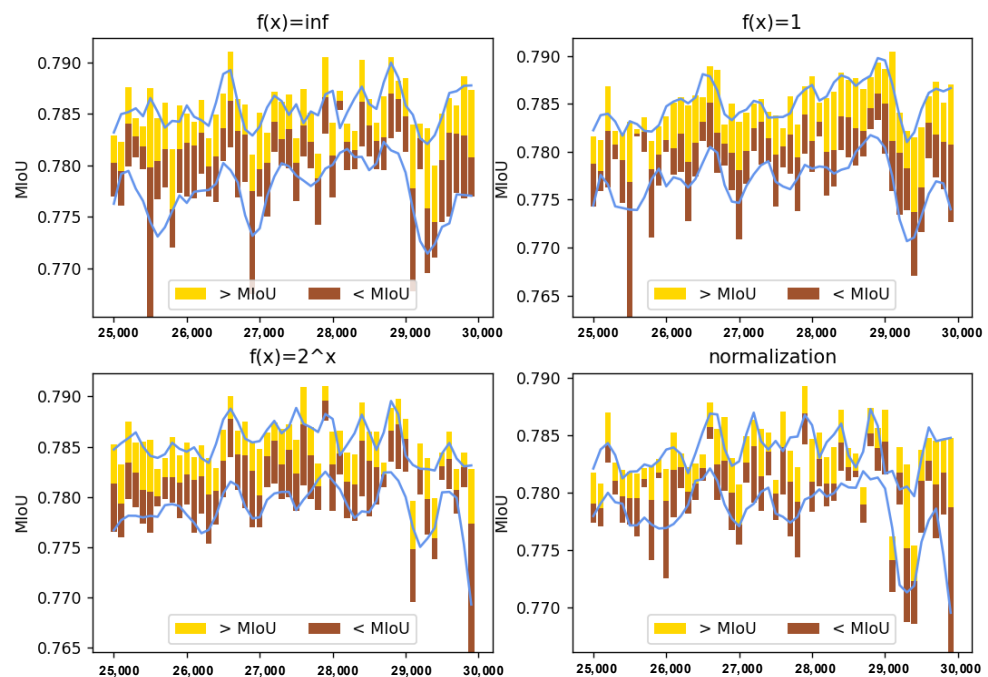


Figure 15. Validation of MIoU for CAPs and PCS and MIoU for normalized weights, where $\inf$ denotes ∞ . The blue lines are the Savitzky-Golay fit lines for the maximum values and the Savitzky-Golay fit lines for the minimum values.

As shown in Figures 13–15, CAPs and the periodic coefficients strategy or normalized weights were used for several experiments to validate the MPA, FWIoU, and MIoU, respectively. In Figures 13 and 14, data fluctuations in MPA and FWIoU have been studied using box plots, with green triangles representing means and green '+' symbols representing outliers. In Figure 15, the real situation of MIoU is validated by taking the iteration interval [25,000: 30,000] and taking the average, [maximum, mean], and [minimum, mean] for the same iteration to highlight the data fluctuations during the training period. The

maximum and minimum values were also smoothed and fitted using the Savitzky–Golay smoothing filter.

3. Overlap Inverse Residual Block

In the OMRB module, the output features have local-to-global information modeled in Overlap convolution's vertical aggregation and MRB's horizontal aggregation, allowing MBA-ASPP to exhibit a significant improvement in image pixel-level classification tasks. Experiments were conducted for baseline and Deeplabv3plus-MBA using conv 1×1 , MRB, Overlap convolution + 3CNN (Overlap convolution followed by a three-layer CNN with 1×1 , 3×3 , 1×1 convolution kernels) and OMRB, respectively, and the experimental data are shown in Table 7. Using conv 1×1 as a basis, for five branches of the baseline, the OMRB module improves the MPA, FWIoU, and MIoU by 0.50%, 0.25%, and 0.21%, respectively; however, for ten branches of the Deeplabv3plus-MBA, the OMRB template improves the MPA, FWIoU, and MIoU by 1.07%, 0.51%, and 0.98%, respectively. The OMRB module's vertical and horizontal aggregation has a significant effect on the feature aggregation of different branches, such as the five-branch ASPP structure in the baseline, which is fundamentally characterized by incomplete feature aggregation but has little effect due to the small number of branches. Deeplabv3plus-MBA's MASPP's ten-branch structure exacerbates branch aggregation incompleteness to some extent due to a large number of branches, but in this study, branch aggregation incompleteness is mitigated to some extent by OMRB's longitudinal and transverse aggregation, resulting in good multi-branch aggregation.

Table 7. Experimental data from baseline and Deeplabv3plus-MBA using Conv 1×1 , MRB, Overlap convolution + 3CNN, and OMRB respectively.

Serial	Baseline				Deeplabv3plus-MBA				MPA	FWIoU	MIoU
	Conv 1×1	MRB	Overlap Convolution + 3 CNN	OMRB	Conv 1×1	MRB	Overlap Convolution + 3 CNN	OMRB			
1	T								87.24	89.68	77.46
2		T							87.99	89.81	77.86
3			T						86.37	89.41	77.00
4				T					87.74	89.93	77.83
5					T				86.89	89.86	78.06
6						T			84.89	88.20	74.39
7							T		87.15	89.99	78.07
8								T	87.96	90.37	79.04

As Table 7 presents experimental data for the semantic segmentation task on the PASCAL VOC 2012 dataset [41], it is not possible to illustrate the effectiveness of the OMRB module for branch aggregation of multi-branch models. Therefore, experiments on image classification were conducted on the CIFAR100 dataset (epoch = 1000, batchsize = 128, optimizer using SGD and setting learning rate $lr = 0.1$ and learning rate strategy using StepLR and setting step_size = 40) using the well-known four-branch model GooLeNet [38] as a benchmark. The experimental data are shown in Table 8. It can be observed that the accuracy of GooLeNet using OMRB is improved by 0.88% compared to conv 1×1 , which also shows the positive effect of OMRB for branching models with multi-layer structures.

Table 8. GooLeNet uses experimental data from conv 1×1 , MRB, Overlap convolution + 3cnn, and OMRB, respectively.

Serial	Conv 1×1	MRB	Overlap Convolution + 3CNN	OMRB	Acc
1	T				56.64
2		T			52.97
3			T		49.20
4				T	57.52

6. Conclusions

This study proposes a multi-branch model with rich branch features and a more thorough branch aggregation based on the ASPP five-branch model, which is not rich enough in branch features and has incomplete branch aggregation. The model uses different methods of extracting features to obtain feature-rich multi-branch modules. Cascaded adaptive parameters and the periodic coefficient strategy are introduced on this basis, allowing the model to adaptively promote or suppress the weight ratios of the branches and weaken the effects of chance uncertainty. The performance of the multi-branch model is further enhanced by using vertical and horizontal aggregation in the branch aggregation phase, which to some extent weakens the phenomenon of incomplete branch aggregation. The validity of the model was verified through a series of experiments on the PASCAL VOC 2012 dataset and the CIFAR100 dataset, which demonstrates the performance of all modules. Further exploratory work could investigate the phenomenon of branching in multi-branch models with local positive and global negative feedback and also in the field of large image segmentation with multi-branch models with multi-layer structures.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, P.C.; Investigation, resources, writing—original draft preparation, visualization, and writing—review and editing, P.C. and D.Y.; supervision, Y.Z. and D.Y.; data curation, R.C., Y.Z., D.Y. and P.C.; project administration and funding acquisition, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a special grant from the Guangdong Provincial Science and Technology Innovation Strategy under Grant No. pdjh2022a0231, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515011326, and program for scientific research start-up funds of Guangdong Ocean University under Grant No. 060302102101.

Data Availability Statement: All datasets utilized in this article are open source and publicly available for researchers to use. Interested individuals can obtain the datasets at the following link: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/> accessed on 13 November 2005. <http://www.cs.toronto.edu/~kriz/cifar.html> accessed on 8 April 2009.

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

References

1. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 386–397. [[CrossRef](#)] [[PubMed](#)]
2. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [[CrossRef](#)]
3. Güler, R.A.; Neverova, N.; Kokkinos, I. DensePose: Dense Human Pose Estimation in the Wild. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7297–7306. [[CrossRef](#)]
4. Zhou, Z.; Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, 20 September 2018; Springer: Cham, Switzerland, 2018; Volume 11045, pp. 3–11. [[CrossRef](#)]
5. Zhou, Z.; Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Trans. Med. Imaging* **2019**, *39*, 1856–1867. [[CrossRef](#)]

6. 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.* **2019**, *43*, 3349–3364. [\[CrossRef\]](#)
7. Berenguel-Baeta, B.; Bermudez-Cameo, J.; Guerrero, J.J. FreDSNet: Joint Monocular Depth and Semantic Segmentation with Fast Fourier Convolutions. *arXiv* **2022**, arXiv:2210.01595.
8. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [\[CrossRef\]](#)
9. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.P.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *40*, 834–848. [\[CrossRef\]](#)
10. Chen, L.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
11. Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; Wang, J. High-Resolution Representations for Labeling Pixels and Regions. *arXiv* **2019**, arXiv:1904.04514.
12. Zhang, T.; Qi, G.; Xiao, B.; Wang, J. Interleaved Group Convolutions. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4383–4392. [\[CrossRef\]](#)
13. Lee, C.; Xie, S.; Gallagher, P.W.; Zhang, Z.; Tu, Z. Deeply-Supervised Nets. *arXiv* **2014**, arXiv:1409.5185.
14. Wang, L.; Lee, C.; Tu, Z.; Lazebnik, S. Training Deeper Convolutional Networks with Deep Supervision. *arXiv* **2015**, arXiv:1505.02496.
15. Wu, H.; Zhang, J.; Huang, K.; Liang, K.; Yu, Y. FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. *arXiv* **2019**, arXiv:1903.11816.
16. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [\[CrossRef\]](#)
17. Yu, F.; Koltun, V.; Funkhouser, T.A. Dilated Residual Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 636–644. [\[CrossRef\]](#)
18. Lin, M.; Chen, Q.; Yan, S. Network in Network. *arXiv* **2014**, arXiv:1312.4400.
19. 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), Las Vegas, NV, USA, 27–30 June 2016; pp. 6230–6239. [\[CrossRef\]](#)
20. Dong, X.; Guan, Y.; Xiaoming, L.; Yang, L.; Jizong, L.; Jing, C.; Qingyu, G. Small sample image classification based on adaptive feature fusion and transformation. *Comput. Eng. Appl.* **2022**, *58*, 223–232. Available online: <https://kns.cnki.net/kcms/detail/11.2127.TP.20210720.1743.011.html> (accessed on 21 July 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 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9992–10002. [\[CrossRef\]](#)
22. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. *arXiv* **2017**, arXiv:1706.03762.
23. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016. [\[CrossRef\]](#)
24. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [\[CrossRef\]](#)
25. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
26. Redmon, J.; Farhadi, A. Yolo3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [\[CrossRef\]](#)
28. Wu, Y.; Liu, Y.; Zhan, X.; Cheng, M. P2T: Pyramid Pooling Transformer for Scene Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 1–12. [\[CrossRef\]](#)
29. Wang, W.; Xie, E.; Li, X.; Fan, D.; 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 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 548–558. [\[CrossRef\]](#)
30. Zhou, K.; Yang, Y.; Cavallaro, A.; Xiang, T. Omni-Scale Feature Learning for Person Re-Identification. *arXiv* **2019**, arXiv:1905.00953v6.
31. Zhou, K.; Xiang, T. Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. *arXiv* **2019**, arXiv:1910.10093.
32. 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. [\[CrossRef\]](#)
33. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M.A. Striving for Simplicity: The All Convolutional Net. *arXiv* **2014**, arXiv:1412.6806.

34. Sen, W.; Kejian, Y. Research and Implementation of Image Scaling Algorithm Based on Bilinear Interpolation. *Autom. Technol. Appl.* **2008**, *7*, 44–45+35. [[CrossRef](#)]
35. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015. [[CrossRef](#)]
36. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773. [[CrossRef](#)]
37. Howard, A.G.; Sandler, M.; Chu, G.; Chen, L.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [[CrossRef](#)]
38. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
39. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472. [[CrossRef](#)]
40. Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv* **2016**, arXiv:1606.08415.
41. Everingham, M.; Eslami, S.M.; Gool, L.V.; Williams, C.K.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2014**, *111*, 98–136. [[CrossRef](#)]
42. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (accessed on 8 April 2009).
43. Deng, J.; Dong, W.; Socher, R.; Li, L.; 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, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.