Multi-Scale Weighted Branch Network for Remote Sensing Image Classification

Kunping Yang, Zicheng Liu, Qikai Lu, Gui-Song Xia*
State Key Lab. LIESMARS, Wuhan University, Wuhan China
{kunpingyang, qikai_lu, guisong.xia}@whu.edu.cn

Abstract

Remote sensing image classification aims to assign semantic label for each input pixel. In this paper, we propose a Multi-Scale Weighted Branch Network (MSWBN) for this dense prediction task. Inspired by attention module, which is commonly adopted to enhance the informative features among the dense feature maps in the deep network, we firstly introduce a Hierarchical Weighted branch Module (HWM). The HWM is designed to extract multi-scale information from the backbone simultaneously with a weighted branches architecture, whose branch weights are generated from lower layers of the backbone. Then, a Low level features Branch Module (LBM) is proposed to embed information with high resolution, where the weighted sum of output from the HWM and low level features is calculated as the dense prediction of the proposed Multi-Scale Weighted Branch Network. The proposed method outperforms extisting best models on the large scale remote sensing image classification dataset (GID) in terms of both efficiency and accuracy.

1. Introduction

Remote sensing image classification is a widely concentrated dense prediction task which can be applied to fields like urban planning, land-use survey and so on. Remote sensing images contain abundant spectral information, based on which many researchers have been focusing on spectral features extraction [28, 34, 32, 6, 37]. Meanwhile, object-oriented methods [43, 41, 42] utilize spatial information to process the prediction. With the remarkable progress achieved by the Convolutional Neural Network (CNN) in computer vision tasks [15, 30, 29, 10, 11, 9, 26], the community has focused on end-to-end dense prediction networks [20, 1, 22, 27, 8, 40, 17, 36, 24, 19, 21, 14, 31] for years. However, how to adapt algorithms to multiscale objects and extract the features fitting different type of input images are still two main challenges we need to face with.

*Corresponding author: guisong.xia@whu.edu.cn.

In this paper, we propose a Multi-Scale Weighted Branch Network architecture to address these two issues in the following aspects.

Firstly, inspired by the parallel multi-scale architecture proposed by several previous works [40, 36, 5, 4, 3], we design a Hierarchical Weighted Branch Module, which contains three hierarchies of dense connceted architecture with multi-scale operations. The DenseASPP module [36] embeds atrous convolutions with different dilation rates in a dense connection way, which covers the feature map in a large scale range with a very dense manner way. Along this way, we build Hierarchical Weighted Branch Module (HWM) with three hierarchies: (1) Each hierarchy consists of several parallel convolutions with not only different dilation rates but also various kernel sizes to extract multi-scale semantic information; (2) We assign each operation with a importance weight which is generated from the lower layers of the chosen backbone, which ensures the proposed module flexible enough to make adjustment according to the input image.

Secondly, we design a Low level features Branch Module (LBM) which connects lower layers with the output of HWM. The shortcut architecture [20, 23, 27, 19, 5] proposed in previous works alleviates the gradient vanishing problem. Moreover, with high resolution information from the lower layers, the whole architecture can retain details like object boundary. However, ignoring the differences between the input images will limit the generalization ability of the algorithm. Based on this insight, we extend the shortcut architecture proposed in pervious works by assigning a branch weight to make a trade-off between the lower level information and the output of HWM.

Different with other works, our proposed Multi-Scale Weighted Branch Network (MSWBN) employs weighted branches architecture to fuse multi-scale information while ensuring the flexibility. Moreover, we analyse the influence of our proposed branch weights on the gradients of the corresponding architecture, base on which we add gates to control each branch weight in HWM and LBM. These gates can help to filtrate the gradients of bad branches. Our main contributions are threefold.

- we propose HWM and LBM that contain several parallel sub-modules to fuse features with various multi-scale information and keep flexibility to make adjustment on the basis of input images.

- We analyse the influence of our proposed branch weights on the gradients of the corresponding architecture and design branch weight gates to make the whole architecture focus on beneficial branches.

- Our proposed method outperforms extisting best models on the large scale remote sensing image classification dataset (GID) in terms of both efficiency and accuracy.

2. Related Work

Our model draws on the success of several areas, including end-to-end dense prediction architectures, multi-scale features extraction, neural architecture search and attention modules.

End-to-end dense prediction architectures. Early works [28, 34, 32, 6, 37] in remote sensing image classification adopt two-stage process, namely features extraction and classification. Besides, some works [43, 41, 42] concentrate on object-oriented process which assigns semantic labels to the preprocessed object segmentation results. Recently, with end-to-end architectures [20, 1, 22, 27, 25, 35] showing significant success in the application to the dense prediction task in computer vision field, researchers begin to employ end-to-end architectures in remote sensing image classification [21, 14, 31]. Due to the characteristics of the dense prediction problem, [20, 1, 22, 27] design derivable decoder modules to extend the output spatial size of the traditional convolutional neural network [15, 30, 29], by which the whole architecture can be end-to-end training and output the dense prediction map directly.

Along this way, we utilize HWM as the encoder module to extract high level features with complex semantic information. Then we propose LBM as the decoder module to fuse the high level features with low level features which contain information with high resolution and more details. The whole network can be trained end-to-end.

Multi-scale features extraction. Identifying multi-scale objects simultaneously is a main challenge we need to face with when dealing with the dense prediction tasks like remote sensing image classification. Hierarchical down-sampling operations always cause the loss of spatial information, which makes multi-scale features extraction intractable. Typically, two types of networks that exploit multiscale features are mainly employed.

First type of networks [20, 23, 27, 19] connect different level features by using the shortcut architecture. [20] employs several shortcuts within the upsampling process and [13] proposes Gated Feedback Refinement Network (G-FRNet) where hierarchical refinements are processed by using the shortcut architecture.

On the other hand, [39, 40, 17, 5, 4, 3] introduce parallel multiscale modules to generate features with different receptive fields simultaneously, with which the whole architecture can accommodate multiscale objects to some degree. As a parallel atrous convolution module, ASPP [3, 4] employs different dilation rates to extract multi-scale information. Moreover, [39, 40, 17, 33] exploit multi-path modules to fuse inputs with different scales.

Along this way, we introduce HWM which contains three hierarchies of dense connceted architecture with multi-scale operations. Each hierarchy can be seen as a parallel multi-scale module. Similar with DenseASPP [36] this design can generate features with more varied receptive fields.

Neural architecture search and Attention modules. Neural architecture search [2, 18] aims to get best model architecture among a potential architecture set.[2] explores the construction of meta-learning techniques for dense image prediction.[18] attains state-of-the-art performance without any ImageNet pretraining.

On the other hand, inspired by SENet [12], many researchers adopt attention modules to enhance the informative features. [7] proposes gated sum to control the information flow which can be seen as a special attention module. [38] designs a module called Attention Refinement Module (ARM) to refine the features of several stages. Pyramid Attention Network (PAN) [16] employs special attention modules where weights map for the lower features are generated by higher features.

In this work, we combine the merits of neural architecture search and attention modules. The branch weights in HWM and LBM control the importance of each branch which can be seen as a soft architecture search to seek the best structure during the training process. Meanwhile, the branch weights are generated from the lower layers of the backbone which can also be seen as a special attention weights map for the corresponding structure. Similar with attention module, this special attention weights can be adjusted to the input images, which ensures the flexibility of the whole architecture.

3. Methods

In this section, we first introduce our proposed Hierarchical Weighted branch Module (HWM). Then the discussion of Low level features Branch Module (LBM) is presented. Finally, we analyse the influence of the branch weights in our proposed modules on the gradients in the training process. The whole algorithm is illustrated in Fig. 1

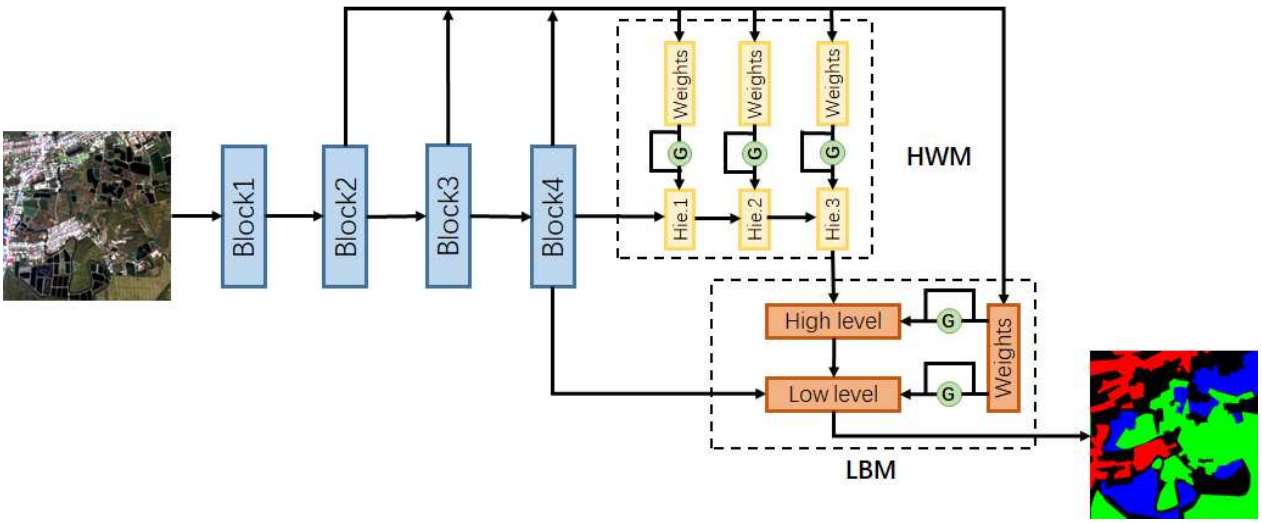

Figure 1. Illustration of our proposed Multi-Scale Weighted Branch Network (MSWBN). We utilize three low layer features from the backbone and Hierarchical Weighted branch Module (HWM) extracts the multi-scale features with scale-sensitive weights. Then Low level features Branch Module (LBM) fuses the low level features with the output of HWM to generate the dense prediction. A weighted sum operation is also applied here. We can choose whether to skip the clip gates in each module.

3.1. Hierarchical Weighted branch Module

Firstly, we discuss our proposed HWM which consists of three hierarchies of parallel multi-scale branches. As illustrated in Fig. 2, these branches contain convolution operations with different kernel sizes and dilation rates.

Concretely, given the output features of the backbone, denoted as F_b, we apply a set of convolution operations $S_o = \{O_{k,r} | k \in \{1,3,5,7\}, r \in \{0,2,4\}\}$ over it, where k denotes the kernel size and r denotes the dilation rate. In implementation, we utilize 8 combinations of k and r, namely, there 8 elements in set S_o. Each operation generates output for one branch. Then, our proposed HWM receives output features from lower layer of the backbone, which we denote as L_1. A convolution operation, denoted as C_1, with kernel size as 3 is applied over L_1 to squeeze the number of channels to 8. We then pass the output of C_1 through a global average pooling layer, denoted as GP, and a softmax function to get the corresponding branch weight, denoted as $W^1 = (W^1_{k1,r1}, ..., W^1_{k8,r8})$, for the first hierarchy.

Consequently, we have:

$$W^1 = softmax(GP(C_1(L_1))) \tag{1}$$

$$F_{h1} = \sum_{k,r} W^1_{k,r} O_{k,r}(F_b) \tag{2}$$

Moreover, the second hierarchy and the third hierarchy receive the output of first hierarchy and second hierarchy as the input and we also apply S_o over them. Similarly, we can get:

$$W^2 = softmax(GP(C_2(L_2))) \tag{3}$$

$$F_{h2} = \sum_{k',r'} W^2_{k',r'} O_{k',r'}(F_{h1}) \tag{4}$$

$$W^3 = softmax(GP(C_3(L_3))) \tag{5}$$

$$F_{h3} = \sum_{k'',r''} W^3_{k'',r''} O_{k'',r''}(F_{h2}) \tag{6}$$

where $W^2_{k',r'}$ and $W^3_{k'',r''}$ are the branch weights for the second hierarchy and third hierarchy respectively and $k', k'' \in \{1,3,5,7\}, r', r'' \in \{0,2,4\}$. C_2 and C_3 are the convolution operations with kernel size of 3, while L_2 and L_3 denote two feature maps from the lower layer.

The proposed HWM can be seen as a weighted graph where connected nodes can be seen as a squential convolution operations. With this design, we can get features which contain information with various scales of receptive fields. For instance, with (1) and (2), we have:

$$F_{h2} = \sum_{k',r'} W^2_{k',r'} O_{k',r'}(\sum_{k,r} W^1_{k,r} O_{k,r}(F_b)) \tag{7}$$

Different operations in S_o can generate features with different receptive fields. Thus, with this compound function form, features that contain information with different combinations of receptive fields are created. Similar with [36], the output receptive field of a sequential two convolution operations can be calculated as $RF_1 + RF_2 - 1$, where RF_i is the receptive field of one convolution operation in the sequential. Given the number of operations in S_o as 8, there are 64 weighted convolution operation sequential with length of two in (4). Thus, F_{h2} contains information with dozens scales of receptive field.

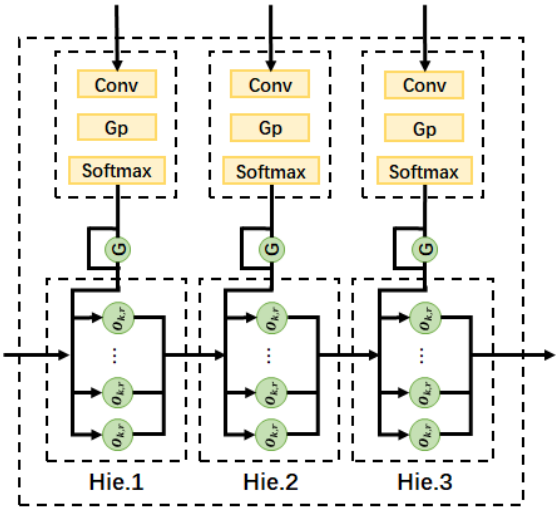

Figure 2. The illustration of Hierarchical Weighted branch Module (HWM). We use three hierarchies of parallel convolution operations to extract features with various scales and each hierarchy receives branch weights from lower layers.

Moreover, the proposed HWM can be approximate to a weighted assembling of 512 sequential convolution operations with length of three, where information with more various scales of receptive field can be extracted simultaneously.

3.2. Low level features Branch Module

In order to fuse features with high resolution, we propose a Low level features Branch Module (LBM) to connect low level features with the output of HWM. As illustrated in Fig. 3, LBM receives the output of HWM and fuses it with features from lower layer of the backbone.

Specifically, LBM contains a weights fusion block and a features fusion block. The weights fusion block receives three branch weights for hierarchies in HWM and generates the weight for the weighted sum operation in features fusion block. Denoting three branch weights for hierarchies in HWM as W^1, W^2 and W^3, the weights fusion block concatenates these weights and applys a Multilayer Perceptron, denoted as MLP, over it. We have:

$$W_{bin} = softmax(MLP(Concat(W^1, W^2, W^3))) \quad (8)$$

where $Concat$ denotes the concatenation operation. $W_{bin} = (W^1_{bin}, W^2_{bin})$ is the weight for the weighted sum operation, which is a vector with length of two, and the $softmax$ function ensures the normalization.

Then, the features fusion block receives F_{h3} and L_3 which has been used to generate the branch weights for the third hierarchy in HWM. We apply two sequences of operations over the F_{h3} and L_3. The kernel size of these convolution operations is 3 and the number of kernels in the last convolution operation of these two sequences is fitted

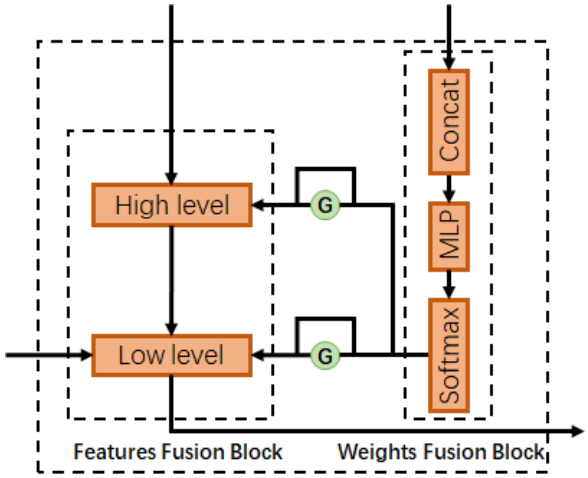

Figure 3. The illustration of Low level features Branch Module (LBM). LBM receives output of HWM and fuses it with low level features by using weighted sum operation.

to the dense prediction problem. Then the weighted sum operation is carried. We have:

$$F_{out} = W^1_{bin} C_H + W^2_{bin} C_L \quad (9)$$

where C_H and C_L are the output of convolution operation sequences applied on F_{h3} and L_3. Finally, we get F_{out} as the output and the loss function is:

$$Loss = \sum_{k,c} Label(k,c) log(F_{s\_out}(k,c)) \quad (10)$$

where $Label(k,c)$ denotes the label of k^{th} pixel for the c^{th} category and $F_{s\_out}(k,c)$ denotes the output value of k^{th} pixel for the c^{th} category in F_{out} after the softmax operation. The loss function is optimized by using stochastic gradient descent (SGD).

3.3. Analyses of grandient and branch weights gate

Analyses of grandient As we mentioned in Sec. 3.1, the proposed HWM can be seen as a weighted graph. In the similar way, the LBM can also be seen as a graph with two edges. The branch weights in these two proposed modules can be seen as the weights for the edges in a graph. These weights control the information flow and decide which branch has more impact on final output. In this section, we analyse how the weights influence the whole architecture. For instance, we analyse the $\frac{\partial Loss}{\partial W^1_{bin}}$. With (9) and (10), we can get:

$$\frac{\partial Loss}{\partial W^1_{bin}} = \sum_{k,c} C_H(k,c)(F_{s\_out}(k,c) - Label(k,c)) \quad (11)$$

Supposing $Label(k,c) = 0$, then we get $F_{s\_out}(k,c) - Label(k,c) > 0$. If $C_H(k,c)$ tends to be large and far above

zero, $\frac{\partial Loss}{\partial W^1_{bin}}$ will be positive and far above zero. If $C_H(k,c)$ tends to be small and far below zero, $\frac{\partial Loss}{\partial W^1_{bin}}$ will be negetive and far below zero. We can get similar result when we suppose $Label(k,c)=1$.

Also, the analyses of the $\frac{\partial Loss}{\partial W^2_{bin}}$ will show the similar result. These analyses mean the weight for the branch in LBM which generates the opposite result to the groundtruth label tends to be smaller during the SGD optimization.

Moreover, we consider the branch weights in HWM and suppose the whole network smooth enough. And then we consider the loss as the function of F_{h1} at position p and has the optimal with respective to it. Thus we can use the second order power series expansion approximation and we have:

$$\begin{aligned} Loss =& Loss(op) + o((F_{h1}(p)-op)^2) \\ & (F_{h1}(p)-op)\frac{\partial Loss}{\partial F_{h1}(p)}(op) + \\ & \frac{1}{2}(F_{h1}(p)-op)^2\frac{\partial^2 Loss}{\partial F_{h1}(p)^2}(op) \end{aligned} \tag{12}$$

where the op denotes the optimal and $o(*)$ represents the higher order infinitesimal. Obviously, $\frac{\partial Loss}{\partial F_{h1}(p)}(op)=0$ and $\frac{\partial^2 Loss}{\partial F_{h1}(p)^2}(op)\geq 0$. With (2) and (12), we have

$$\frac{\partial Loss}{\partial W^1_{k,r}} \approx \sum_p C_p^{k,r}(F_{h1}(p)-op)\frac{\partial^2 Loss}{\partial F^2_{h1}(p)}(op) \tag{13}$$

where $C_p^{k,r}$ is the value of $O_{k,r}(F_b)$ at position p which is a constant with respective to $F_{h1}(p)$. With (13), we know that if $F_{h1}(p) > op$ and $O_{k,r}(F_b)$ tends to be larger than zero, $\frac{\partial Loss}{\partial W^1_{k,r}}$ will tend to be larger than zero. If $F_{h1}(p) < op$ and $O_{k,r}(F_b)$ tends to be smaller than zero, $\frac{\partial Loss}{\partial W^1_{k,r}}$ will tend to be larger than zero. We can get similar result with respect to $W^2_{k,r}$ and $W^3_{k,r}$.

These analyses also show that weight for the branch which generates the opposite result to the groundtruth label tends to be smaller during the SGD optimization. The characteristic we get above ensures the whole architecture focuses on the branch which is more suitable when given input images.

Branch weights gate Based on this insight we choose relu gate to control the gradient flow if the weight is below the threhold. Specifically, in HWM we have:

$$cl(weight) = \begin{cases} \theta & weight < \theta \\ weight & weight \geq \theta \end{cases} \tag{14}$$

In LBM, to avoid the small output tensor, we times clip gate by 2. And then, we have:

$$\frac{\partial Loss}{\partial C_H(k,c)} = 2cl(W^1_{bin})(F_{s\_out}(k,c) - Label(k,c)) \tag{15}$$

$$\frac{\partial Loss}{\partial C_p^{k,r}} \approx cl(W^1_{k,r})(F_{h1}(p)-op)\frac{\partial^2 Loss}{\partial F^2_{h1}(p)}(op) \tag{16}$$

$$\frac{\partial Loss}{\partial *} = \frac{\partial Loss}{\partial cl(W^1_{bin})}\frac{\partial cl(W^1_{bin})}{\partial *} \tag{17}$$

$$\frac{\partial Loss}{\partial *} = \frac{\partial Loss}{\partial cl(W^1_{k,r})}\frac{\partial cl(W^1_{k,r})}{\partial *} \tag{18}$$

The $*$ in (17) and (18) denotes the element which is used to generate W^1_{bin} and $W^1_{k,r}$ respectively. With (15) and (16), we know that we can control the gradient flow for the branches by setting the θ in clip gate. And then, with (17) and (18), we can get that the gradient flow of the weight generators which belong to unsuitable branches tend to collapse to zero and make no impact on the whole architecture while the weights become smaller.

4. Experimental Analysis

In order to study the effectiveness and efficiency of our proposed method, we evaluate it on Gaofen Image Dataset (GID) [33] which is a large scale remote sensing image classification dataset. We first slice the images in GID into patches with size of 512×512. Then we select resnet101 as our backbone and study the effectiveness of each module we propose in terms of the output accuracy and training speed. Also we compare the proposed method with some existing best models and our method achieves the state-of-the-art. Firstly, we study how the number of channels of operations in HWM influences the output result and training speed. Then, we study the influence of clip gate in HWM and LBM on the output accuracy. Futhermore, to verify the adaptive capacity of our proposed architecture, we resample the image patches in GID and evaluate models on an extremely label-imbalanced dataset.

In details, we choose resnet101 as the backbone and set initial learning rate as 0.007. We employ 'poly' policy with power of 0.9. We train the network on Titan X with memory of 12G for 20 epochs. We use stochastic gradient descent to update parameter and set the batch size as 10, momentum as 0.9 and weight decay as 0.0001. As the same as deeplabv3plus, we use random flip and random scale between 0.5 and 2 as the data augmentation. We set θ in HWM and LBM as 0.125 and 0.5 respectively.

4.1. Gaofen Image Dataset

GID consists of 150 GF-2 satellite images which cover more than 70,000 km^2. Widespread over several cities in China and containing rich geographic information including morphology characters and so on, GID presents 5 land cover categories, which is built-up, farmland, forest, meadow, and water, and a background category. The background category represents the unknown area which can not be identified artificially. In this work, we slice the images

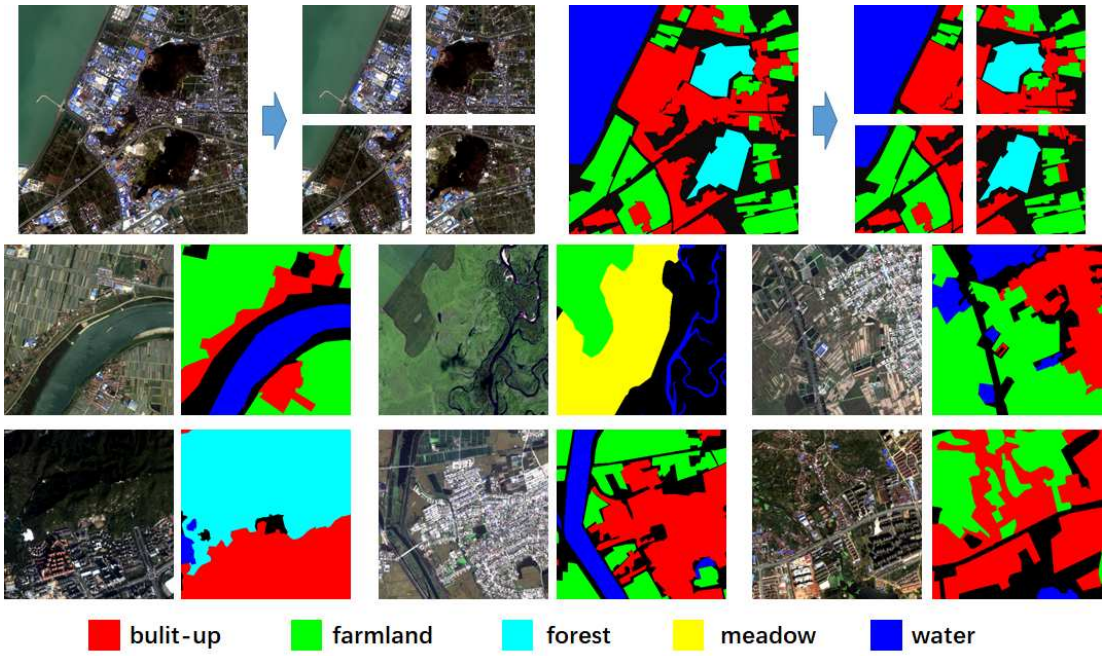

Figure 4. The illustration of patches of GID. The first row shows how we generate the patches GID dataset. The second and third row show examples of patches dataset.

in GID into patches with size of 512×512 to fit the architecture we propose. The examples of patches in GID can be seen in Fig. 4. Same with [33], we first evaluate modules we introduce by utilizing 120 training images and 30 testing images, which are sliced into 22560 training patches and 5640 testing patches. Then we compare our method and existing best models on testing patches GID images. Finally, the resampling is processed to generate a extremely label-imbalanced dataset where the adaptive capacity of the dense prediction models can be verified.

4.2. Number of channels of operations in HWM

Firstly, we study the influence of the operations in HWM. In implementation, we choose $S_o = \{O_{k,r}|k \in \{1,3,5,7\}, r \in \{0,2,4\}\}$ containing 8 different operations which are $O_{1,0}$, $O_{3,0}$, $O_{5,0}$, $O_{5,2}$, $O_{5,4}$, $O_{7,0}$, $O_{7,2}$ and $O_{7,4}$. These convolution operations with different kernel sizes and dilation rates can extract features with different scales. Meanwhile, the number of channels of these convolution operations influences the model capacity. Generally, models with inadequate model capacity will suffer underfitting and fail to acquire complex conception corresponding to the concerned problem. On the other hand, models with overmuch model capacity will suffer over-fitting and lose the model generalization.

In this section, we set the number of channels of convolution operations as 64, 96, 128, and 160. With this setting, we can seek for the appropriate model capacity for this dense prediction problem. The result is shown in Tab.1.We

Method	mIOU	OA	training speed
64 channels	90.13	96.34	0.7467 sec/step
96 channels	89.79	96.27	0.7967 sec/step
128 channels	**90.49**	**96.40**	0.8188 sec/step
160 channels	90.24	96.05	0.8725 sec/step
Deeplabv3plus	89.87	96.17	0.9603 sec/step

Table 1. The evaluation of HWM containing operations with different numbers of channels on testing patches GID dataset. mIOU represents mean Intersection over Union and OA represents overall accuracy.

can see that operations in HWM with 128 channels attain a appropriate model capacity and whole architecture outperforms deeplabv3plus. Training speed decreases with the number of channels increasing. It is worth mentioning that our best model only needs about 85% of training time compared with deeplabv3plus, which mainly because of the fewer parameters in weighted sum operation compared with concatenation + convolution.

4.3. Clip gates

As we analyse in Sec.3.3, theoretically, the weights for branches which tend to generate opposite output to the groundtruth label are getting smaller during the SGD optimization. We utilize this characteristic to process a soft neural architecture search which reduces the influence of bad parts of the whole architecture instead of removing them. Moreover, we propose clip gates to control the gradient flow

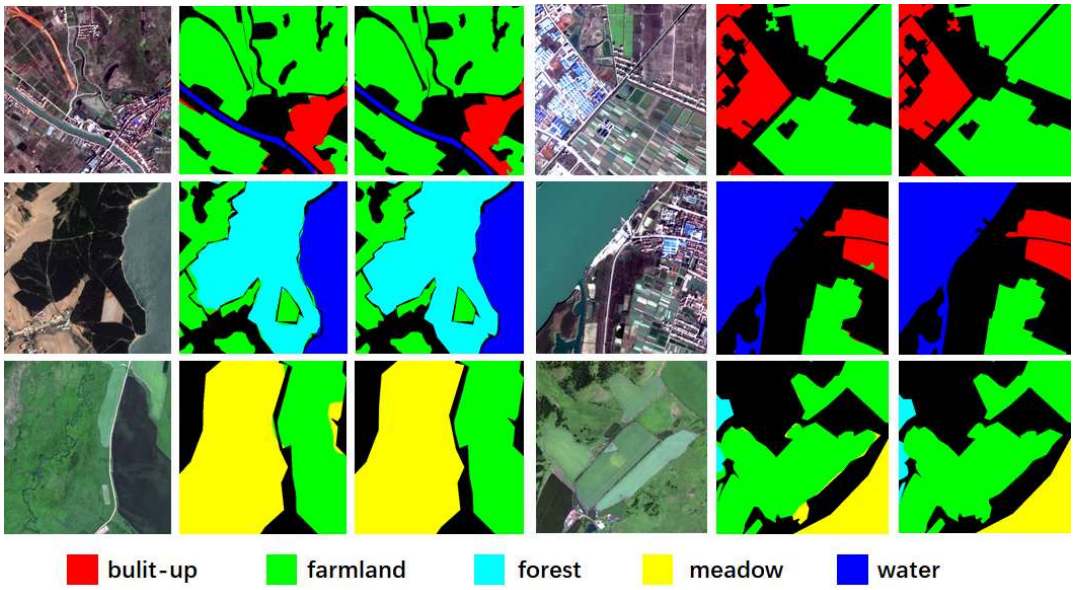

Figure 5. Visualization results on patches GID dataset when employing our best model. The first and forth cols are the origin patches images. The second and fifth cols are the dense prediction we generate. The third and sixth cols are the groundtruth

which can reduce the impact made by bad parts of the whole architecture. In this section, we do ablation experiment to study the experimental influence of the clip gates in HWM and LBM. We first remove the clip gates in HWM and LBM respectively and then we wipe out the clip gates in the whole architecture. With these settings, we can figure out how the clip gates in each module influence the output result. The result is shown in Tab.2, Tab.3 and Tab.4.

Method	mIOU	OA
64 channels-no gates in HWM	89.71	96.18
96 channels-no gates in HWM	88.78	95.92
128 channels-no gates in HWM	89.10	96.06
160 channels-no gates in HWM	89.93	96.16

Table 2. The evaluation of the MSWBN without clip gates in HWM on testing patches GID dataset.

Method	mIOU	OA
64 channels-no gates in LBM	89.53	96.14
96 channels-no gates in LBM	89.45	96.03
128 channels-no gates in LBM	89.95	95.99
160 channels-no gates in LBM	89.83	96.15

Table 3. The evaluation of the MSWBN without clip gates in LBM on testing patches GID dataset.

We can see that removing clip gates causes accuracy loss in varying degrees compared with Tab.1 which verifies the advantage of clip gates and supports the analyses in Sec.3.3.

Method	mIOU	OA
64 channels-no gates	89.40	96.04
96 channels-no gates	88.84	95.99
128 channels-no gates	89.21	95.99
160 channels-no gates	88.40	95.61

Table 4. The evaluation of the MSWBN without clip gates in the whole architecture on testing patches GID dataset.

4.4. Comparing with state-of-the-art methods

In this section, we compare our proposed methods with several extisting state-of-the-art algorithms including deeplabv3plus and object-oriented method proposed in [33] on testing patches GID dataset. Different with method proposed in [33], our MSWBN and deeplabv3plus are end-to-end. The result is shown in Tab.5

Method	mIOU	OA
MSWBN	**90.49**	**96.40**
Deeplabv3plus	89.87	96.17
Object-oriented method in [33]	87.25	95.76

Table 5. The evaluation of the MSWBN on testing patches GID dataset compared with existing best methods.

We can see that our proposed MSWBN achieves the best result which outperforms deeplabv3plus by 0.62 in mIOU and 0.23 in OA. And also, MSWBN outperforms method proposed in [33] by 3.24 in mIOU and 0.64 in OA.

4.5. Resampling experiment

To verify the adaptive capacity of the proposed architecture, we resample the patches we used before and generate a extremely label-imbalanced patches dataset. Generally, imbalanced dataset will cause the model collapse and ignore the underprivileged categories. Supposing there are 99% of farmland pixels and 1% of built-up pixels, the model which monotonously predicts farmland category will get a overall accuracy of 99%. This collapsed model is useless in actual production. Benefited from flexible branch weights which can adjust the architecture with respective to the input images, the method we propose can learn which branch weight should be larger while the input image contains underprivileged categories. The statistical distribution of the patches dataset we used before and the resampled extremely label-imbalanced patches dataset can be viewed in Tab. 6.

	built-up	framland	water	forest	meadow
original train	143‰	571‰	173‰	69‰	45‰
original test	243‰	431‰	260‰	61‰	4‰
resampled train	168‰	538‰	181‰	68‰	45‰
resampled test	150‰	490‰	285‰	76‰	**0.7‰**

Table 6. The statistical distribution of the original and resampled patches GID dataset

We can see that the meadow category in resampled test dataset is extremely underprivileged which challenges the adaptive capacity of models.

We set $S_o = \{O_{1,0}, O_{3,0}, O_{5,0}, O_{5,2}, O_{5,4}\}$ which contains only 5 convolution operations with different kernel sizes and dilation rates in this experiment. We modify the θ to 0.2 in HWM and evaluate with mIOU metrics which concerns the underprivileged categories. The experimental result is shown in Tab.7

Method	mIOU
5 operations with no LBM	87.28
5 operations with LBM	87.99
5 operations with LBM and clip gates	**89.09**
deeplabv3plus	84.57

Table 7. The evaluation of the MSWBN on resampled patches GID dataset.

We can see that our proposed architecture keeps the accuracy while the dataset is extremely label-imbalanced.

5. Conclusion

In this work, we propose a MSWBN consisting of HWM and LBM. HWM extracts multi-scale features simultaneously through three hierarchies of weighted parallel convolution operations. LBM fuses the low layer information to the output dense prediction. Moreover, we analyse the gradient of branch weights in HWM and LBM based on which we propose the clip gate to control the gradient flow. Then, we evaluate the proposed method on GID dataset. The experiment results show that our proposed MSWBN outperforms existing best models and the clip gates do improve the output accuracy which verifies our analyses of the gradients in each module.

References

[1] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *CoRR*, abs/1505.07293, 2015. 1, 2

[2] Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8713–8724, 2018. 2

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 1, 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 1, 2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[6] Begüm Demir and Lorenzo Bruzzone. Histogram-based attribute profiles for classification of very high resolution remote sensing images. *IEEE Trans. Geoscience and Remote Sensing*, 54(4):2096–2107, 2016. 1, 2

[7] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[8] Jun Fu, Jing Liu, Yuhang Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *CoRR*, abs/1708.04943, 2017. 1

[9] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448, 2015. 1

[10] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for

visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015. 1

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7132–7141, 2018. 2

[13] Md. Amirul Islam, Mrigank Rochan, Neil D. B. Bruce, and Yang Wang. Gated feedback refinement network for dense image labeling. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4877–4885, 2017. 2

[14] Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. Learning aerial image segmentation from online maps. *IEEE Trans. Geoscience and Remote Sensing*, 55(11):6054–6068, 2017. 1, 2

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012. 1, 2

[16] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid Attention Network for Semantic Segmentation. *arXiv e-prints*, page arXiv:1805.10180, May 2018. 2

[17] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5168–5177, 2017. 1, 2

[18] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L. Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. *CoRR*, abs/1901.02985, 2019. 2

[19] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. Parsenet: Looking wider to see better. *CoRR*, abs/1506.04579, 2015. 1, 2

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[21] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geoscience and Remote Sensing*, 55(2):645–657, 2017. 1, 2

[22] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1520–1528, 2015. 1, 2

[23] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *CoRR*, abs/1606.02147, 2016. 1, 2

[24] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters - improve semantic segmentation by global convolutional network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1743–1751, 2017. 1

[25] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3309–3318, 2017. 2

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 1

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 234–241, 2015. 1, 2

[28] Qian Shi, Xiaoping Liu, and Xin Huang. An active relearning framework for remote sensing image classification. *IEEE Trans. Geoscience and Remote Sensing*, 56(6):3468–3486, 2018. 1, 2

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[31] Yiting Tao, Miaozhong Xu, Fan Zhang, Bo Du, and Liangpei Zhang. Unsupervised-restricted deconvolutional neural network for very high resolution remote-sensing image classification. *IEEE Trans. Geoscience and Remote Sensing*, 55(12):6805–6823, 2017. 1, 2

[32] Piotr Tokarczyk, Jan Dirk Wegner, Stefan Walk, and Konrad Schindler. Features, color spaces, and boosting: New insights on semantic classification of remote sensing images. *IEEE Trans. Geoscience and Remote Sensing*, 53(1):280–295, 2015. 1, 2

[33] Xin-Yi Tong, Qikai Lu, Gui-Song Xia, and Liangpei Zhang. Large-scale land cover classification in gaofen-2 satellite imagery. In *2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018*, pages 3599–3602, 2018. 2, 5, 6, 7

[34] Devis Tuia, Michele Volpi, Mauro Dalla Mura, Alain Rakotomamonjy, and Rémi Flamary. Automatic feature learning for spatio-spectral image classification with sparse SVM. *IEEE Trans. Geoscience and Remote Sensing*, 52(10):6062–6074, 2014. 1, 2

[35] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison W. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 1451–1460, 2018. 2

[36] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes.

In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3684–3692, 2018. 1, 2, 3

[37] Xiwen Yao, Junwei Han, Gong Cheng, Xueming Qian, and Lei Guo. Semantic annotation of high-resolution satellite images via weakly supervised learning. *IEEE Trans. Geoscience and Remote Sensing*, 54(6):3660–3671, 2016. 1, 2

[38] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pages 334–349, 2018. 2

[39] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, pages 418–434, 2018. 2

[40] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239, 2017. 1, 2

[41] Ji Zhao, Yanfei Zhong, and Liangpei Zhang. Detail-preserving smoothing classifier based on conditional random fields for high spatial resolution remote sensing imagery. *IEEE Trans. Geoscience and Remote Sensing*, 53(5):2440–2452, 2015. 1, 2

[42] Yanfei Zhong, Rongrong Gao, and Liangpei Zhang. Multiscale and multifeature normalized cut segmentation for high spatial resolution remote sensing imagery. *IEEE Trans. Geoscience and Remote Sensing*, 54(10):6061–6075, 2016. 1, 2

[43] Yanfei Zhong, Ji Zhao, and Liangpei Zhang. A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery. *IEEE Trans. Geoscience and Remote Sensing*, 52(11):7023–7037, 2014. 1, 2