Integrating Gate and Attention Modules for High-Resolution Image Semantic Segmentation

Zixian Zheng, Xueliang Zhang, *Member, IEEE*, Pengfeng Xiao, *Senior Member, IEEE*, and Zhenshi Li

***Abstract*—Semantic segmentation of high-resolution (HR) remote sensing images achieved great progress by utilizing deep convolutional neural networks (DCNNs) in recent years. However, the decrease of resolution in the feature map of DCNNs brings about the loss of spatial information and thus leads to the blurring of object boundary and misclassification of small objects. In addition, the class imbalance and the high diversity of geographic objects in HR images exacerbate the performance. To deal with the above problems, we proposed an end-to-end DCNN network named GAMNet to balance the contradiction between global semantic information and local details. An integration of attention and gate module (GAM) is specially designed to simultaneously realize multiscale feature extraction and boundary recovery. The integration module can be inserted in an encoder-decoder network with skip connection. Meanwhile, a composite loss function is designed to achieve deep supervision of GAM by adding an auxiliary loss, which can help improve the effectiveness of the integration module. The performance of GAMNet is quantitatively evaluated on the ISPRS 2-D semantic labeling datasets and achieves state-of-the-art performance in comparison with other representative methods.**

***Index Terms*—Attention module (AM), gate module (GM), high-resolution (HR) remote sensing imagery, semantic segmentation.**

I. Introduction

SEMANTIC segmentation of high-resolution (HR) remote sensing images received lots of attention in recent years due to the development of deep convolutional neural networks (DCNNs) [1]–[6]. The goal of semantic segmentation is to assign a semantic label to every pixel in an image [7]. It has a wide range of applications that can be roughly divided into two categories. One is to label a single category, such as road extraction [8], [9], building segmentation [10], [11], ship detection [12], cloud segmentation [13], [14], and water area segmentation [15]. The other is to label multiple categories all together [16]–[20]. However, DCNN-based semantic segmentation of HR images still faces the challenges of the high diversity of geographic objects and the class imbalance originating from the high-spatial resolution [21], [22]. High diversity often presents high intra-class heterogeneity and low inter-class variance, such as the confusion between trees and low vegetation, as well as the confusion of manmade objects in urban areas [23]. Class imbalance refers to that different objects take different area ratios in an image, which leads to the uneven distribution of the categories.

To solve these challenging problems, DCNN models are specifically improved for semantic segmentation of HR images. For the semantic segmentation task, DCNN evolves as an end-to-end structure by jointly learning feature extraction from original input data to final output with great generalization capabilities [24], [25]. With the benefit of the consecutive down-sampling operations in convolutional forward stage, the extracted features are containing rich context and being useful for object segmentation. However, it leads to the loss of detailed information that is crucial for accurate segmentation of small objects and boundaries. Therefore, considering the characteristics of HR images and the trade-off between semantic context and detail information, mainly two types of improvements on DCNNs have been explored for semantic segmentation of HR images [26], [27]: one type of making full use of multiscale features and the other type of enhancing boundary information.

For semantic segmentation task of HR images, objects of different sizes need feature maps with different field-of-view because of different semantic context in different levels of feature maps. Lower-level feature maps support the location of small objects and higher-level feature maps guarantee the context for complete large objects. Hence, for an image with huge diversity, it is important to make full use of the multiscale features from the encoder part. Simple cascading or summing operations cannot effectively solve the above problems well, and may bring redundancy of feature information. Generally, there are three ways to capture multiscale features, including share-net, pyramid pooling net, and skip-net [28], as shown in Fig. 1.

Share-net has a shared deep network with differently resized inputs as shown in Fig. 1(a). It resizes the original image into several different scales and puts each into the shared network. A fusion module is applied to learn the multiscale features from each branch. Chen *et al.* [28] and Yang and Peng [29] proposed a share-net network with two input scales aided by an attention model. In remote sensing community, Lin *et al.* [30] proposed two branches of fully convolutional network with different input scales for maritime semantic labelling. It is noted that only two

Manuscript received December 24, 2020; revised February 21, 2021; accepted April 1, 2021. Date of publication April 6, 2021; date of current version May 13, 2021. This work was supported in part by the National Science and Technology Major Project of China under Grant 21-Y20A06-9001-17/18, in part by the National Natural Science Foundation of China under Grant 42071297, Grant 41871235, and Grant 41871326, in part by the Fundamental Research Funds for the Central Universities under Grant 020914380080, and in part by the High-level Innovation and Entrepreneurship Talents Introduction Program of Jiangsu Province of China. *(Corresponding author: Xueliang Zhang.)*

The authors are with the Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Key Laboratory for Land Satellite Remote Sensing Applications of Ministry of Natural Resources, School of Geography and Ocean Science, Nanjing University, Nanjing 210023, China (e-mail: zhengzx95@gmail.com; zxl@nju.edu.cn; xiaopf@nju.edu.cn; lzhenshi@outlook.com).

Digital Object Identifier 10.1109/JSTARS.2021.3071353

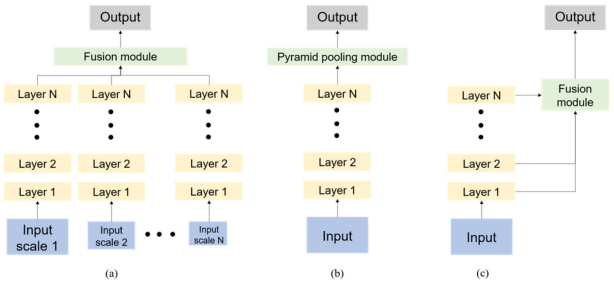


Fig. 1. Three types of networks for capturing multiscale features. (a) Share-net. (b) Pyramid pooling net. (c) Skip-net.

input scales were used in the above studies. In theory, we can add more branches in a share-net to capture multiscale features better, but it is limited by the computation capacity and the training time.

Pyramid pooling net designs filters or pooling operations at multiple effective field-of-views to process the high-level features, which can generate multiscale high-level features [31], [32] as shown in Fig. 1(b). The pyramid pooling module in PSPNet [31] generates four-scale high level features by four pooling operations. The atrous spatial pyramid pooling (ASPP) module in DeepLabV3+ [32] functions similarly by four filters with different dilated rates. Several works of remote sensing image semantic segmentation [12], [33] also adopted the pyramid pooling module. No matter what operation is adopted, the pyramid pooling net can effectively enlarge the field-of-view and make full use of the high-level features to reveal multiscale context. However, a lack of low-level features would lead to missing details of objects [34], [35].

Skip-net combines multilevel features from the intermediate layers as shown in Fig. 1(c). Since the high-level features contain more semantic information and the low-level features explain more details, a fusion of features at different levels is valuable to distinguish objects with various appearances in terms of both spectra and geometry. SegNet [34] and RefineNet [35] are typical skip-net structures which take use of the intermediate layers for segmentation. Skip-net is a commonly used structure for remote sensing image semantic segmentation to aggregate multilevel features [23], [36]. However, not all the features are proved to be effective and overmuch fusion of the intermediate layers would lead to information redundancy, thus it's of great importance to select the effective features for different objects.

Several structures focused on enhancing boundary information to deal with the blurring of boundary detail caused by the pooling or convolution operations in DCNNs. By recovering detailed boundaries, some confusion among manmade objects can be relieved, such as the confusion between cars and roads. Dilated convolution [37] was introduced to replace convolutional layer and pooling layer. It can enlarge the receptive field without reduction of resolution and thus can keep more details around boundary. The skip connection in the encoder part can also do good to recover details [34], [35], [38]. In addition, the edge detection method is combined with deep network to provide explicit boundary information. One type of this method detects the boundary and then classifies in the deep learning architecture [39], [40]. For example, Fu *et al*. [40] proposed an object-based deep CNN framework that integrates object-based image analysis with deep CNNs to accurately extract and estimate impervious surfaces. The other type integrates edge detection in a deep learning network [27], [41]. For example, Marmanis *et al*. [42] proposed a boundary-aware network by training boundary probability maps with holistically nested edge detection to support semantic segmentation.

In this article, we propose a novel end-to-end deep convolutional skip-net. On the basis of a symmetrical encoder-decoder structure, we design a data-driving integration module which can be inserted in the decoder part with skip connection to the encoder part. The integration module takes multiscale features from the encoder part as input to fully exploit and learn multiscale information. It integrates an attention module (AM) and a gate module (GM), which can extract features of different scales for different objects to deal with the problems of high diversity and class imbalance, and at the same time make full use of the information that is conducive to boundary segmentation in low-level features to improve the accuracy of boundary segmentation. The integration module is a plug-and-play module which can be easily inserted into an encoder-decoder structure. The main contributions of this article are concluded as followed.

1) A novel end-to-end network structure (GAMNet) for HR image semantic segmentation is proposed by inserting an integration module into a skip-net to realize multiscale feature selection and boundary detail recovery simultaneously.
2) The data-driving plug-and-play integration module (GAM) can be easily inserted in encoder-decoder networks, which is demonstrated to be effective for improving segmentation accuracies.
3) The proposed GAMNet is demonstrated achieving state-of-the-art performance through a large number of experiments, especially for the small objects (e.g., the category of car) on the ISPRS two-dimensional (2-D) semantic segmentation benchmarks.

II. Methods

To deal with the challenges of boundary blurring and the tradeoff between semantic context and local details, we design a novel end-to-end network called GAMNet on the basis of an encoder-decoder structure for excavating multilevel features. GAM (integration of gate and attention module) is inserted into the encoder-decoder network to accomplish both the selection of useful features and the optimization of boundary.

A. Overview

The overall framework adopts the encoder-decoder structure as illustrated in Fig. 2. In the encoder part, convolutional layers and pooling layers gradually reduce the spatial resolution and expand the receptive field to extract semantic information. The decoder part gradually improves the spatial resolution to restore the detail information. By keeping the symmetry of encoding and

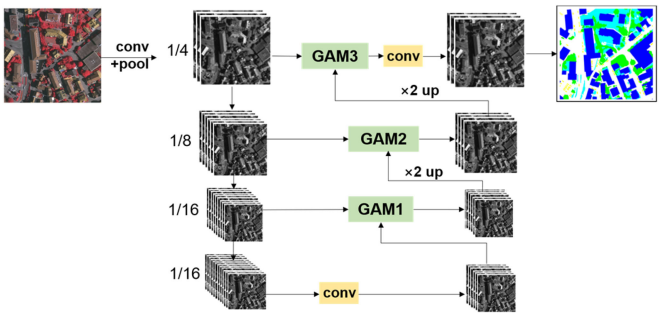


Fig. 2. Overall framework of the proposed GAMNet structure.

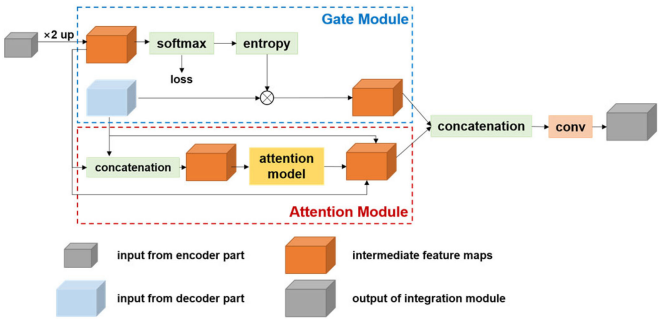


Fig. 3. Structure of the integration of gate and attention module (GAM).

decoding structure, the contradiction between global semantic information and local detailed information is balanced.

In the encoder part of the network, we take ResNet-101 as the baseline. ResNet [43], [44] has been proved as an effective convolutional network structure to extract image features. The basic unit of ResNet-101 is the bottleneck building block. According to the spatial resolution of feature map, we divided ResNet-101 into four ResNet blocks, as shown in the encoder part of Fig. 2. Each ResNet block is composed of different number of bottlenecks. The dilated convolution [45], also called as atrous convolution, is introduced in the fourth ResNet block to enlarge the receptive field without decreasing the spatial resolution, and thus to alleviate the contradiction between the global semantic information and the local detail information to some extent. The size of the dilated convolution kernel is controlled by the dilated convolution rate. A larger rate brings a wider receptive field but may also cause the discontinuity of the feature map.

In the decoder part of the network, we design a plug-and-play GAM module and insert it into a skip-net to fuse multiscale features. Three GAMs are inserted in the skip-net to aggregate features of ResNet blocks in encoder part from high-level to low-level, named GAM1, GAM2, and GAM3. GAM is composed of an AM and a GM, taking the features at a high level from the encoder and the features at the adjacent lower level from the decoder as input. The detailed structure of a GAM is illustrated in Fig. 3.

B. Attention Module

The feature maps at each scale in the encoder part are useful, but features at different scales have different importance to various objects. Hence, it is of great importance to make an optimum selection of features for the very heterogeneous objects in HR images. The attention model can be used to measure the importance of different receptive fields to different features [46]. It was primarily used in natural language processing [47] and then introduced to computer vision for image classification [48], object detection [49], and semantic segmentation [28], [50]. Attention model can be divided into soft and hard attention [51]. Among the existing DCNNs with attention model, the soft attention is differentiable, and can thus be used to compute the gradient through the neural network and to learn the weights of features by propagating forward and backward [52].

Therefore, we apply a hard attention model to weight features at different scales, aiming at realizing the optimization of multi-scale features for various objects. Exactly, attention mechanisms can be learned by adding an additional feed-forward propagation to the network to measure the importance of features at different scales. The multiscale features can then be optimized effectively by a weighted fusion separately for each pixel.

The structure of the AM in GAM is shown in the bottom half of Fig. 3. The attention model measures the feature importance of different scales according to the prior information by setting the two convolutional layer structure, restraining the invalid features, and retaining the effective features. The input of the AM includes feature maps from the encoder part and up-sampled feature maps at the adjacent higher level in the decoder part. The two input feature maps are concatenated and filtered by two convolutional layers to evaluate the scale importance. Let x represent the pixel in feature maps, ω $(k \times k \times 2)$ represent the weight of each pixel from different scales, S represent the number of scales (here $S = 2$), s represent a single scale ($s = 1$ or 2), f $(k \times k \times d_1)$ represent the multiscale input feature maps from encoder and decoder parts, h $(k \times k \times d_2)$ represent the output feature maps of the two-layer AM, and A $(k \times k \times d_3)$ represent the weighed summation of multiscale features. The input (f) and output (A) of the AM are expressed as follows:

$$A_x = \sum_{s=1}^{S} \omega_x^s \cdot f_x^s \ , \ \text{where} \ \omega_x^s = \frac{\exp\left(h_x^s\right)}{\sum_{t=1}^{S} \exp\left(h_x^t\right)} \ . \quad (1)$$

In the AM, the weights of different scales are first calculated and then the weighted summation is calculated as the final output. Through the AM, the importance of features at different scales is separately measured for each pixel. The feature maps are fused according to the weighed importance to realize effective optimization of multiscale features.

C. Gate Module

With the deepening convolutional layers, the high-level feature maps tend to lose local detail information, which often results in fuzzy boundaries in segmentation result even though the decoder part is applied for recovering details. To deal with the problem of blurring boundaries, we apply a GM [53] to recover boundary details effectively by fully excavating the detailed low-level features. GM works as a function to control the flow of information in DCNN-based semantic segmentation tasks. Different gate functions have been designed for different tasks. Islam *et al.*[54] designed gate units and gated refinement units

to weigh high-level features for controlling low-level features. Li *et al.* [55] proposed a gated fully fusion module which can fuse features by comparing the feature maps to enhance the dissemination of useful information.

In this article, we take the information entropy as a gate function to impose boundary constraints on feature maps. Usually, the misclassification probability is higher near the boundaries and thus the corresponding information entropy in these areas is higher. In addition, the detailed information is urgently needed for these areas to accurately localize the boundaries. Therefore, the information entropy is used as the gate function to obtain more low-level feature information around object boundaries, which serves as a boundary constraint to better recover the detailed information.

The structure of the GM is shown in the top part of Fig. 3. The input is the same as that of the AM, including up-sampled high-level feature maps in the decoder part and adjacent low-level feature maps in the encoder part. The gate calculation is conducted on the up-sampled high-level feature maps, which is assigned as a weight value to the low-level feature maps. After fusing by multiplying the weight and the low-level feature maps, the output of the GM is generated. Let $F(k \times k \times d_4)$ represent the output, f^{high} $(k \times k \times d_5)$ represent the high-level feature maps, and f^{low} $(k \times k \times d_6)$ represent the low-level feature maps. The input and output of the GM are expressed as follows:

$$F = \left(G\left[\text{Softmax}\left(f^{\text{high}} \times \omega_{1*1}\right)\right] \dot{\times} f^{\text{low}}\right) + f^{\text{high}}, \quad (2)$$

where G represents the formula of information entropy gate function, as shown in (3). ω_{1*1} represents 1×1 convolutional layer, "$\times$" and "$\dot{\times}$" stand for matrix multiplication and element-wise multiplication, respectively,

$$G\ (x) = \ -\sum_{i=1}^{N} p_i\,(x) \log_2\,(p_i\,(x)), \quad (3)$$

where N represents the number of categories, x represents the pixels in an image and $p_i(x)$ represents the probability of each land cover category.

The high-level features are calculated by the Softmax layer to indicate the probability of each land cover category. Accordingly, the probability vector has the same number of channels for each pixel and can thus be used to calculate the information entropy, which is used as a gate to control the number of low-level features for different parts. In this way, the low-level features are processed under the guidance of high-level features through the information entropy in the GM, aiming at better recovering the missing detail information near the boundaries and thus improving the semantic segmentation accuracies.

D. Integration Module (GAM)

On the basis of the encoder-decoder with skip connection, an integration module which can be easily inserted is proposed for improving the performance. The integration module is formulated by combing the GM and AM to screen multi-scale features and to restore boundary details simultaneously. To take advantage of both AM and GM to accomplish fusing useful features and optimizing boundaries, we design a novel integration module combining AM and GM. A concatenation operation is designed to fuse the output of AM and GM, followed by a convolutional layer next to it, as shown in Fig. 3. The concatenation operation preserves the characteristics from both modules very well and a convolution with a kernel of 3×3 is employed for better understanding the hybrid features.

In addition, an effective loss function plays a very important role for training the proposed model. To prevent the gradient from disappearing within the network, the auxiliary loss was proposed to combine with the original loss function, which can result in easier optimization. For example, Trinh *et al.* [56] proposed an auxiliary loss to improve the generalization ability of recurrent neural network. Zhao *et al.* [31] employed an auxiliary loss for ResNet optimization, which was proved to be effective for improving performance.

Accordingly, we specially design a composite loss for the proposed network to achieve easier optimization as well as to make GAM work better. Specifically, an auxiliary cross entropy loss from the GM is combined with the widely used cross entropy loss. The weight parameters λ are set to balance the two types of loss. The composite loss is defined as follows:

$$\mathcal{L} \ = \mathcal{L}_{\text{cel}} \ + \lambda \mathcal{L}_{\text{GM}} \quad (4)$$

where $\mathcal{L}_{\text{cel}}$ represents the cross entropy loss, $\mathcal{L}_{\text{GM}}$ represents the cross entropy loss from GM, which is the summation of three GMs. The weight parameter λ is set as 0.5 by default, and its influence on the segmentation results will be discussed in Section IV-B.

III. Experiments

A. Dataset and Preprocessing

The proposed GAMNet method is evaluated on the two open benchmarks of ISPRS 2-D semantic labelling challenge, i.e., Vaihingen[1] and Potsdam[2] dataset. Both the datasets are airborne images, consisting of high resolution true ortho photo (TOP) tiles and corresponding digital surface models (DSMs) derived from dense image matching techniques. We only use the near infrared, red, and green bands for both training and testing. Each dataset has been manually labeled into six land cover categories, including impervious surfaces (ImSurface), buildings (building), low vegetation (LowVeg), trees (tree), cars (car), and clutter/background (clutter). The clutter/background class includes water bodies and other objects that look very different from everything else. The ground truth of all tiles in both datasets was released. In this article, the eroded references are used for evaluation to be consistent with the methods on ISPRS 2-D semantic labelling WEBSITE,[3] where the boundaries of objects are eroded by a circular disc of three-pixel radius.

1) Vaihingen Dataset: Vaihingen is a relatively small village with many detached buildings and small multistory buildings.

[1]Online. [Available]: http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html

[2]Online. [Available]: http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

[3]Online. [Available]: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

The Vaihingen dataset contains 33 tiles of different sizes TOP images. It includes the near infrared, red, and green bands delivered by the camera. The average size of the tiles is 2494 × 2064 pixels with a spatial resolution of 9 cm. Similar to previous studies, we use 16 tiles for training (ID: 1, 3, 5, 7, 11, 13, 15, 17, 21, 23, 26, 28, 30, 32, 34, 37) and the rest 17 tiles are for testing [52], [53].

2) Potsdam Dataset: Potsdam shows a typical historic city with large building blocks, narrow streets, and dense settlement structure. The Potsdam dataset contains 38 tiles of the same size 6000 × 6000 pixels with a spatial resolution of 5 cm. Similar to previous studies, we use 23 tiles for training and 14 tiles for testing the model. The tiles 7–10 are not used because of the labelling mistakes.

3) Data Preprocessing: Due to the limitation of GPU memory, we crop the Vaihingen and Potsdam datasets into patches of 512 × 512 pixels with a stride of 100 pixels and 400 pixels, respectively. To avoid overfitting, we implement data augmentation (DA) to the cropped patches by flipping each patch horizontally and vertically and rotating each patch 90° counter-clockwise.

B. Implementation Details

1) Computing Environment: The proposed method is implemented using the TensorFlow platform. All the experiments are performed in the equipment with a GeForce RTX 2080Ti GPU.

2) Evaluation Metrics: In order to comprehensively evaluate the performance of the semantic segmentation results, we chose the overall accuracy (*OA*), *F1*-score (*F*1) and intersection over union (*IoU*) as evaluation metrics. The formulas of these metrics are as follows:

$$OA = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{fp} + \mathrm{fn}} \tag{5}$$

$$\mathrm{recall} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \tag{6}$$

$$\mathrm{precision} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \tag{7}$$

$$F1 = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{8}$$

$$IoU = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}, \tag{9}$$

where tp , tn, fp, and fn represent true positives, true negatives, false positives, and false negatives, respectively. In addition, *MF1* and *MIoU* are calculated as the mean *F*1 and the mean IoU of each category (without the category of clutter/background), respectively.

3) Overlay Inference (OI): Due to the limitation of memory, we also crop the images into patches in the inference stage. However, this causes inconsistent segmentations across the borders of the patches. In order to mitigate the boundary effect, the OI strategy is employed. The overlapped patches are firstly sliced by setting an overlay ratio. The probability of land cover category is averaged for each pixel in overlapped areas. As shown in Fig. 4, compared with inferencing without overlap, the overlapped area in the blue box can make use of information from all its surrounding patches. Hence, overlapped areas can aggregate more context information for inferencing by this overlay strategy, the effectiveness of which will be proved in the following Section IV-C.

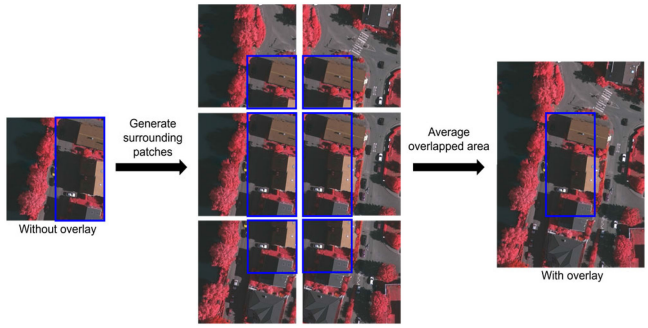


Fig. 4. Diagram of OI by setting the overlay ratio as 50%, where the blue box represents the overlapped area and the parts outside the blue box represent the context information that can be used for inferencing the overlapped area.

4) Design of Experiments: The weights of ResNet-101 in the encoder part are pretrained on ImageNet [59]. The "poly" learning rate policy is adopted with the initial learning rate setting as 0.001. The batch size of all the experiments is 4 and the network is trained for 50 epochs.

In order to demonstrate the effectiveness of the proposed GAMNet, we design the following sets of experiments.

1) Comparing the network trained with or without GAM to clearly reveal the effectiveness of the integration module.
2) Comparing the network trained with different encoders to demonstrate the effectiveness of the plug-and-play module.
3) Comparing the network trained by the composite loss with that trained by only cross entropy loss to demonstrate the function of the composite loss.
4) Effects of patch size for both training and inference are discussed together with the OI, aiming at demonstrating the importance of context information for multiclass semantic segmentation by DCNN.
5) Proposed GAMNet is compared with previous representative methods on the two open benchmarks of ISPRS 2-D Semantic Labelling Challenge to validate its effectiveness.

IV. Results

A. Evaluation of the Proposed GAMNet Method

Altogether, we introduced three strategies to enhance the GAMNet performance, including multigrid (MG), DA, and OI. MG refers to employing dilated convolution into ResNet. We set the dilated rate as 1, 2, and 4 in the fourth block of ResNet for the bottlenecks respectively. DA is also used in the inference stage in addition to the training stage. Five different scales {0.5, 0.75, 1, 1.25, 1.5} and left-right flipping counterpart are input to the trained network for inferencing and averaged as the final output. In terms of OI, the overlay ratio is set as 50% for this experiment. Table I gives the evaluation metrics of the GAMNet results on the Vaihingen dataset. The original GAMNet achieves a *MF1* of

TABLE I
ACCURACIES OF GAMNET RESULTS WITH AND WITHOUT ENHANCING BY THE STRATEGIES OF MG, DA, AND OI ON THE VAIHINGEN TEST SET

Model	*MG*	*DA*	*OI*	*MF*1 (%)	*MIoU* (%)	*OA* (%)
GAMNet				88.83	80.21	90.18
	√			88.89	80.32	90.28
	√	√		90.04	82.13	91.02
	√	√	√	**90.20**	**82.38**	**91.11**

TABLE II
COMPARISON OF THE ACCURACIES FOR DIFFERENT NETWORK STRUCTURES ON THE VAIHINGEN TEST SET. THE NETWORKS ARE BASELINE, WITH AM, WITH GM, AND WITH HYBRID GATE AND ATTENTION MODULE (GAM) FROM THE FIRST TO THE FOURTH ROW. THE SIZE OF THE TEST IMAGE PATCHES IS 512 × 512

AM	GM	*F*1 (%)					*MF*1 (%)	*MIoU* (%)	*OA* (%)	Model size (m)	Test time (ms)
		ImSurface	Building	LowVeg	Tree	Car					
		89.85	94.34	81.16	87.89	60.90	82.83	72.26	88.31	43.26	32
√		91.69	94.87	82.82	89.20	81.97	88.11	79.11	89.81	54.09	37
	√	91.67	94.89	82.61	89.20	80.81	87.83	78.71	89.75	45.83	36
√	√	**92.24**	**95.20**	**83.22**	**89.29**	**84.19**	**88.83**	**80.21**	**90.18**	56.13	40

88.83%, *MIoU* of 80.21% and *OA* of 90.18%. It is shown that all the strategies for enhancing are effective. MG, DA, and OI successively improve the performance of *MF1* by 0.1%, 1.1%, and 0.2%, respectively. After enhancing, the *MF1*, *MIoU*, and *OA* of GAMNet results are improved to 90.20%, 82.38% and 91.11%, respectively.

B. Ablation Study for the Integration Module

In the proposed GAMNet, the integration module aims to simultaneously accomplish the selection of useful features and the optimization of boundaries. To verify the effectiveness of the integration module, the networks with GM, AM, and GAM are compared together with the baseline on the Vaihingen dataset, where the baseline has the structure with ResNet-101 in the encoder part and taking bilinear interpolation as the up-sampling methods.

As given in Table II, when only AM is used, the *MF1* increases to 88.11% which is 5.28% higher than that of the baseline. Similarly, GM achieves 87.83% *MF1* by an improvement of 5.0% compared with the baseline. Moreover, the performance of the GAM is further boosted up. It obtains higher accuracies compared with AM and GM and improves about 6.0% of *MF1* over the baseline. According to the comparison results, the integration module is proved to be effective from the perspective of accuracy. In addition, the parameters of model size and test time are selected for comparison on computation complexity. Compared with AM, GM, and baseline, GAMNet has the highest model size of approximately 56 million parameters. Furthermore, the test time for these four networks is similar with difference less than 8 ms.

To further show the effectiveness of the integration module, the *F*1 of each category as well as the segmentation results on the Vaihingen test set are presented. Tile 27 in the test set is taken as an example to show the segmentation results as in Fig. 5. The top row in Fig. 5 presents the original inference result and the bottom two rows show the partially enlarged drawing. From the second row in Fig. 5, we can see that GM performs better in boundary constraint than AM, as for the building in the upper right in pink box, even though such difference does not change the *F*1 apparently. It is illustrated in Table II that AM can achieve 0.28% higher *MF1* than GM. However, according to the *F*1 of each category, only the *F*1 for car of AM is 1.16% higher than that of GM, and the *F*1 of the remaining four categories are about the same. In the bottom row of Fig. 5, many cars are orderly distributed. The baseline can hardly distinguish the difference between cars. Both GM and AM achieve better segmentation of cars compared with the baseline. However, GM tends to segment the adjacent cars into a mixed object, while AM is able to distinguish the single cars. Combining the findings from the second and the third rows, we can know that AM and GM play the complementary roles. AM can select and fuse multi-scale features, which achieves better segmentation for various objects in an image, especially for the small objects of cars. GM has the ability to constrain the boundary through information entropy and thus achieves more accurate building boundaries. Accordingly, the GAM achieves higher accuracy in terms of each category compared with only AM or GM, demonstrating the effectiveness of the integration GAM module.

The outputs of GM and AM modules are combined by concatenation followed by a convolutional layer. To verify the effectiveness of the proposed combination strategy, it is compared with other combination strategies, including summation, summation followed by a convolutional layer, and concatenation only, as given in Table III. The fusion operation of summation and concatenation deliver almost the same quality in terms of *MF1* score and *MIoU*. The 3 × 3 convolutional layer following the concatenation can improve the *MF1* by 0.21%, which demonstrates the superiority of the proposed strategy of combining concatenation and convolutional layer.

The integration module is effective when taking ResNet-101 as encoders. To further verify the improvement of the plug-and-play module, ResNet-50 and ResNet-152 are selected as encoders for comparison with ResNet-101. As given in Table IV, all of the networks with different encoders deliver good performance with similar quality in accuracies, which demonstrates

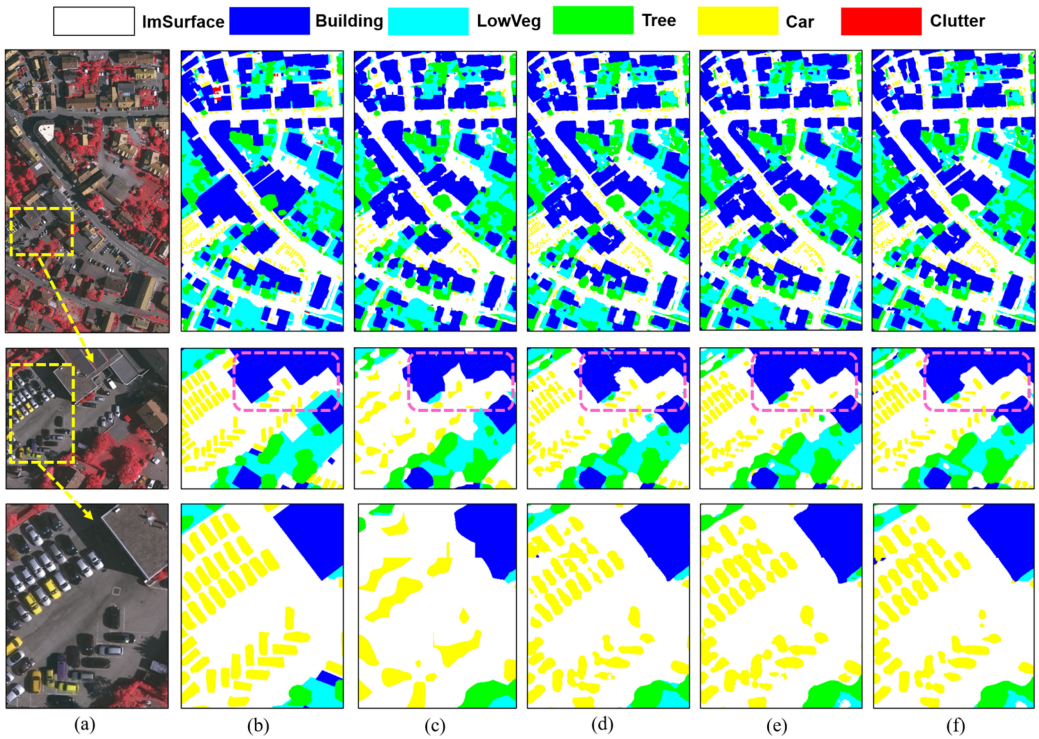


Fig. 5. Comparison of the semantic segmentation results from different network structures on Vaihingen tile 27, where AM, GM, and GAM refer to the network with AM, with GM, and with hybrid GAM, respectively. (a) Image. (b) Ground truth. (c) Baseline. (d) AM. (e) GM. (f) GAM.

TABLE III
COMPARISON OF GAMNET ACCURACIES BY USING DIFFERENT COMBINATION METHODS ON THE VAIHINGEN TEST SET, WHERE CONV REPRESENTS CONVOLUTIONAL LAYER

Model	Combination method	*MF*1 (%)	*MIoU* (%)
GAMNet	sum	88.63	79.90
	concatenate	88.59	79.85
	sum+conv	88.61	79.90
	concatenate+conv	**88.83**	**80.21**

TABLE IV
COMPARISON OF THE ACCURACIES FOR DIFFERENT ENCODERS ON THE VAIHINGEN TEST SET

Encoder	*F*1 (%)					*MF*1 (%)	*OA* (%)
	ImSurface	Building	LowVeg	Tree	Car		
ResNet-50	92.14	94.84	83.22	89.44	83.53	88.63	90.11
ResNet-101	92.37	95.22	83.21	89.42	84.25	88.89	90.28
ResNet-152	92.59	95.33	83.31	89.43	84.27	88.99	90.38

that GAM is effective to be inserted in an encoder-decoder network.

The composite loss function is proposed to make the network easier to be optimized. In addition, the inference result with and without the auxiliary loss are compared to see whether the composite loss can improve the segmentation accuracies. As given in Table V, where CEL represents training only with cross entropy loss and CPL represents training with composite loss as described in Section II-D, CPL outperforms CEL by approximately 0.7% in *MF*1 and 1.0% in *MIoU*. CPL significantly improves the accuracy for the category of car by 2.3% in *F*1 compared with CEL, and achieves an average improvement by about 0.3% for other categories.

A region with irregularly distributed cars in Vaihingen tile 4 is zoomed in and shown in Fig. 6 to take a further look at the improvement of car segmentation by CPL, where the gray box and pink box give an intuitive sense of the difference. For example, the two cars in the upper left of the pink box are sticking by CEL because of misclassifying the shadow between cars, but are separated by CPL. Only a fraction of the two cars in the lower right of the pink box are identified by CEL compared to

TABLE V
COMPARISON OF SEGMENTATION ACCURACIES BY USING DIFFERENT LOSS FUNCTIONS FOR TRAINING GAMNET ON THE VAIHINGEN TEST SET, WHERE CEL REFERS TO CROSS ENTROPY LOSS AND CPL REFERS TO COMPOSITE LOSS

Loss	*F*1 (%)					*MF*1 (%)	*MIoU* (%)
	ImSurface	Building	LowVeg	Tree	Car		
CEL	91.94	94.97	82.89	89.37	82.01	88.24	79.32
CPL	**92.37**	**95.22**	**83.21**	**89.42**	**84.25**	**88.89**	**80.32**

TABLE VI
COMPARISON OF ACCURACIES OF GAMNET TRAINED BY DIFFERENT PATCH SIZES ON THE VAIHINGEN TEST SET

Patch size	*F*1 (%)					*MF*1 (%)	*MIoU* (%)	*OA* (%)
	ImSurface	Building	LowVeg	Tree	Car			
256×256	92.00	95.21	84.40	89.93	84.74	89.26	80.86	90.48
512×512	**92.94**	**95.58**	**84.47**	**90.28**	**86.95**	**90.04**	**82.13**	**91.02**

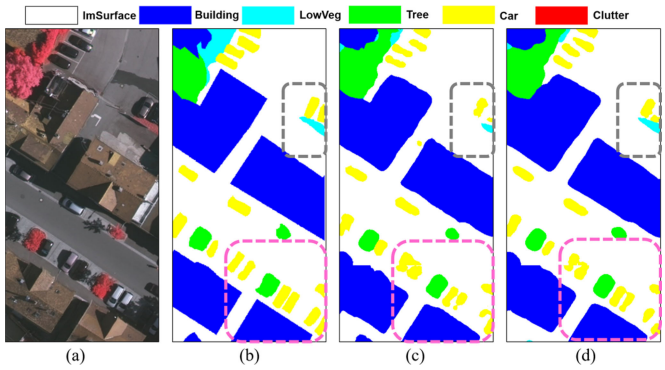


Fig. 6. Comparison of segmentation results in Vaihingen tile 4 produced by GAMNet with different loss functions, where CEL refers to cross entropy loss and CPL refers to composite loss. (a) Image. (b) Ground truth. (c) CEL. (d) CPL.

CPL. Hence, the quantitative and qualitative evaluation results prove that CPL is able to improve the segmentation performance effectively compared with CEL.

Above experiments are performed by setting λ as 0.5. In order to further understand the influence of λ on the training performance, we set the value of λ ranging from 0.4 to 1 (in interval of 0.1). According to the evaluation on both the Vaihingen and Potsdam datasets, the accuracy changes caused by setting different λ are within 0.3% in *MF*1. For the Vaihingen dataset, the best result is obtained when λ is set as 0.9. For the Potsdam dataset, the best performance is achieved by setting λ as 0.8.

C. Effect of Patch Size

Due to the limitation of memory, large-size remote sensing images are usually cropped into patches for both training and inferencing. In theory, a larger patch is assumed to contain more complete objects and more spatial context information for inferencing, which is beneficial to semantic segmentation. In this section, we set different patch sizes in the training stage and the inferencing stage respectively to validate the influence of patch size.

The influence of patch size on training is firstly evaluated. Considering the balance between the calculation complexity and object completeness, we only compare patch size of 256 × 256 pixels and 512 × 512 pixels. A set of training patches with 256 × 256 pixels in a 128 stride is prepared for comparison. It should be noted that we set the same inferencing patch size as that of training here. Table VI gives the performance of patch size 256 × 256 pixels and 512 × 512 pixels with the aid of MG and DA strategies. Patch size of 512 × 512 pixels outperforms patch size of 256 × 256 pixels by 0.7% in *MF1* and 1.2% in *MIoU*. This shows that with the increase of patch size, the integrity of objects in patches is improved and more contextual information can be used, which leads to a better performance.

To visually illustrate the influence of training patch size, an example inference result in Vaihingen tile 2 is shown in Fig. 7, where the top row is the original inference result of tile 2 and the bottom row is the partially enlarged drawing. As marked by the pink rectangles in the top row, the GAMNet trained using patches with 512 × 512 pixels obviously gives a more complete segmentation result. Some buildings are missing by the network trained using patches with 256 × 256 pixels but are completely segmented by patches with 512 × 512 pixels. As illustrated in the bottom row, cars parked next to trees are missing and misclassified by patches with 256 × 256 pixels compared to the result from patches with 512 × 512 pixels which is consistent with the ground truth.

The influence of patch size for inference is evaluated together with OI, where the GAMNet is trained by patches with 512 × 512 pixels. On one hand, the OI can help to mitigate the boundary effect caused by the cropped patches with inconsistent segmentation across the borders. On the other hand, the overlay strategy can aggregate more context information for the overlapped areas, which is similar to the function of enlarged inferencing patches, as presented in Fig. 4. In order to reveal the influence of the overlay ratio, we set the overlay ratio from 0%, 25%, 50%, to 75% for each inferencing patch size, respectively. In addition, we set inferencing patch size as 256 × 256 pixels, 512 × 512 pixels, 1024 × 1024 pixels, and 1536 × 1536 pixels respectively to make combination with OI. The evaluation results for different inferencing patch sizes together with different overlay ratios are given in Table VII, through which we can learn the influence of each as well as the interaction between them. According to the results in Table VII, it can be concluded as: with the increase of inferencing patch size, segmentation accuracies tend to be improved for each

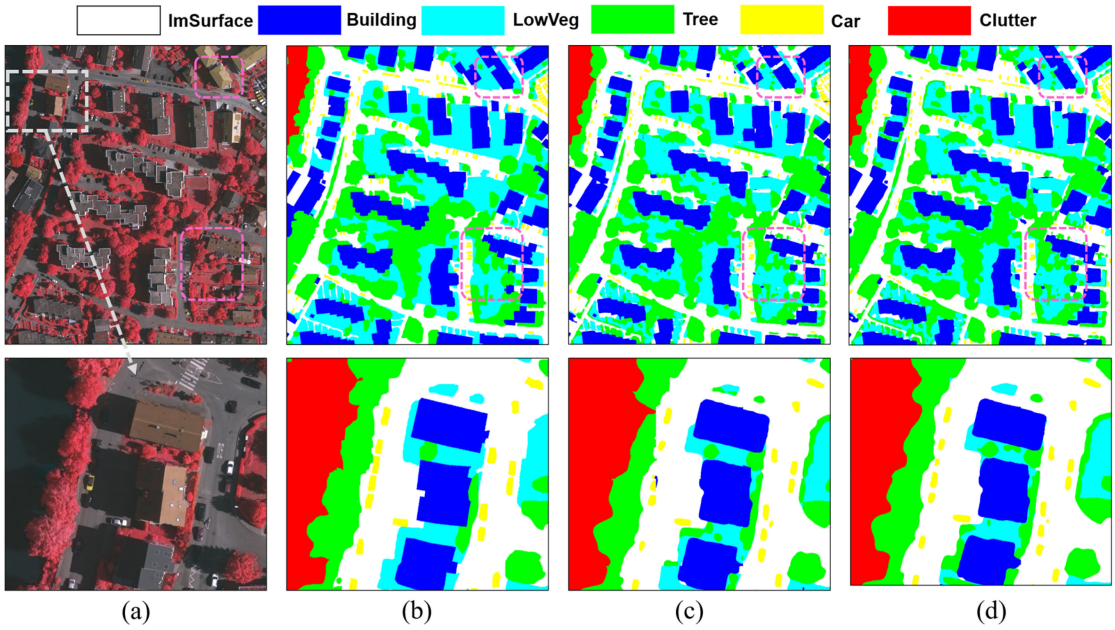


Fig. 7. Comparison of segmentation results in Vaihingen tile 2 produced by GAMNet trained with different patch sizes. (a) Image. (b) Ground truth. (c) Patch 256. (d) Patch 512.

TABLE VII
COMPARISON OF ACCURACIES OF GAMNET TRAINED BY DIFFERENT PATCH SIZES ON THE VAIHINGEN TEST SET

Overlay ratio (%)	*MF*1 (%)			
	256×256	512×512	1024×1024	1536×1536
0	88.23	88.89	89.25	89.18
25	89.11	89.28	89.40	89.45
50	89.08	89.26	89.38	89.49
75	89.33	89.39	89.37	89.49

TABLE VIII
ACCURACIES OF EACH CATEGORY IN GAMNET RESULTS WITH DIFFERENT PATCH SIZES FOR INFERENCING ON THE VAIHINGEN TEST SET

Patch size	*F*1 (%)					*MF*1 (%)	*OA* (%)
	ImSurface	Building	LowVeg	Tree	Car		
256×256	91.76	94.54	82.68	89.12	83.07	88.23	89.72
512×512	92.37	95.22	83.21	89.42	84.25	88.89	90.28
1024×1024	92.54	95.45	83.53	**89.57**	**85.14**	**89.25**	90.50
1536×1536	**92.59**	**95.46**	**83.55**	89.56	84.73	89.18	**90.51**

overlay ratio; the accuracies of results with overlay outperform those without overlay strategy, because all the ratios can help improve segmentation accuracies compared with no OI. However, for different inference patch size, the best performance comes from different overlay ratios; and as the inferencing patch size increases, the difference between the result with overlay and that without overlay tends to be smaller. In summary, we can know that both the increase of inferencing patch size and the OI strategy help improve the semantic segmentation performance. There is a mutual interaction between them since both the strategies function to expand spatial context information for inferencing. Accordingly, the larger the inferencing patch size, the smaller the effect of OI on the performance improvement.

To further show the effect of inferencing patch size on multiclass semantic segmentation, we take a deep look at the *F*1 for each of the five categories. The GAMNet trained on patches with 512 × 512 pixels is used for comparison without overlay inferencing strategy. As given in Table VIII, as the inferencing patch size increases, the *F*1 for each category tends to be improved with a declining growth rate. It is noted that the categories of car and tree achieve the highest *F*1 in the patch size of 1024 × 1024 pixels while the other categories achieve the best performance when the patch size for inference is 1536 × 1536 pixels. Compared with the other categories, the average size of cars is relatively smaller, which could be the reason why the best inferencing patch size for cars is relatively smaller than that for other categories.

TABLE IX
QUANTITATIVE COMPARISON BETWEEN GAMNET AND OTHER METHODS ON THE ISPRS VAIHINGEN TEST SET

Model	*F*1 (%)					*MF*1 (%)	*OA* (%)
	ImSurface	Building	LowVeg	Tree	Car		
SVL_3	86.6	91.0	77.0	85.0	55.6	79.0	84.8
ADL_3	89.5	93.2	82.3	88.2	63.3	83.3	88.0
UZ_1	89.2	92.5	81.6	86.9	57.3	81.5	87.3
ONE_7	91.0	94.5	84.4	89.9	77.8	87.5	89.8
DLR_10	92.3	95.2	84.1	90.0	79.3	88.2	90.3
CASIA2	**93.2**	**96.0**	84.7	89.9	86.7	90.1	91.1
GSN3	92.2	95.1	83.7	89.9	82.4	88.7	90.3
DP-DCN	92.2	95.6	79.9	89.8	74.6	86.4	89.2
TreeUNet	92.5	94.9	83.6	89.6	85.9	89.3	90.4
UFMG_4	91.1	94.5	82.9	88.8	81.3	87.7	89.4
PDM	91.5	94.7	81.9	88.5	74.0	86.1	89.2
GAMNet	93.0	95.7	**85.1**	**90.5**	**88.5**	**90.6**	**91.3**

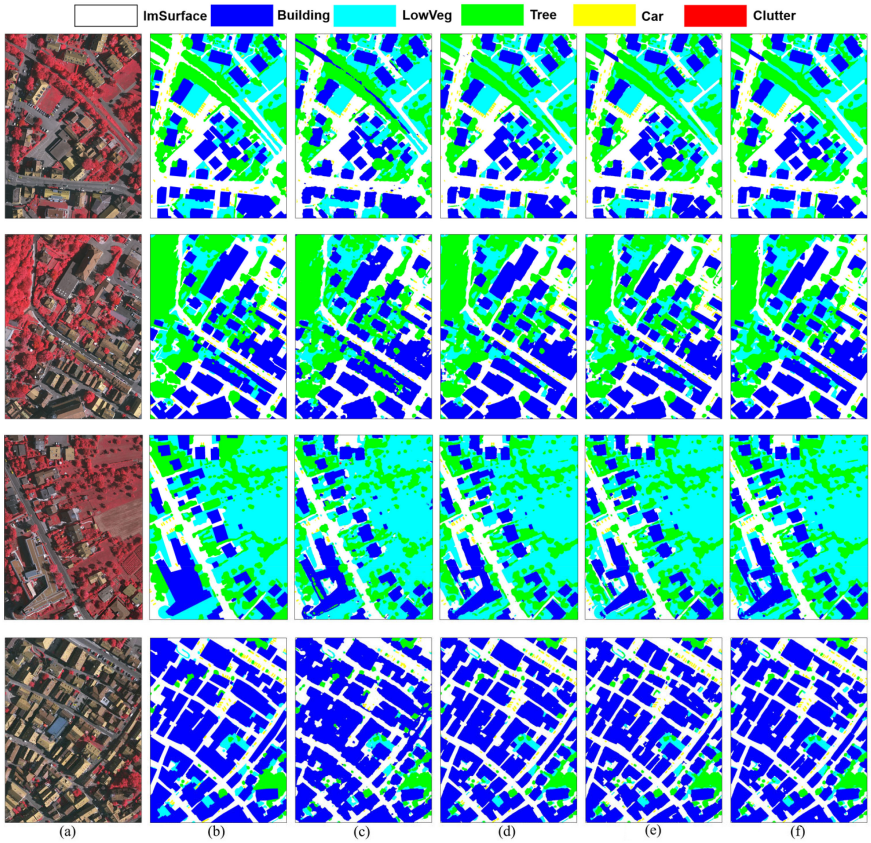


Fig. 8. Visual comparison between GAMNet and other methods on Vaihingen tile 8, 6, 4, and 24. (a) Image. (b) Ground truth. (c) UZ_1. (d) CASIA2. (e) GSN3. (f) GAMNet.

D. Comparison on Benchmark Sets

The effectiveness of the proposed GAMNet is further validated by comparing with representative results submitted to ISPRS committee or in published research papers.

The comparison is first performed on the ISPRS Vaihingen dataset. It should be noted that we only use near infrared, red, and green bands for training our GAMNet network. The GAMNet network is trained with patch size 512 × 512 pixels and the parameters λ of the composite loss is set to 0.9. The inference is performed by patches with 512 × 512 pixels with 75% overlay ratio considering the balance between time efficiency and evaluation performance. The methods involved for comparison include the following.

TABLE X
QUANTITATIVE COMPARISON BETWEEN GAMNET AND OTHER METHODS ON THE ISPRS POTSDAM TEST SET

Model	*F*1 (%)					*MF*1 (%)	*OA* (%)
	ImSurface	Building	LowVeg	Tree	Car		
SVL_3	84.0	89.8	72.0	59.0	69.8	74.9	77.2
UZ_1	89.3	95.4	81.8	80.5	86.5	86.7	85.8
CASIA2	93.3	97.0	87.7	88.4	96.2	92.5	91.1
SWJ_2	94.4	97.4	87.8	87.6	94.7	92.4	**91.7**
TreeUNet	93.1	97.3	86.8	87.1	95.8	92.0	90.7
UFMG_4	90.8	95.6	84.4	84.3	92.4	89.5	87.9
HUSTW4	93.6	**97.6**	**88.5**	88.8	94.6	92.6	91.6
MCA	**94.7**	92.9	83.2	88.9	84.3	88.8	90.1
ResUNet-a	93.5	97.2	88.2	**89.2**	96.4	92.9	91.5
GAMNet	93.8	97.5	88.4	89.0	**96.6**	**93.1**	**91.7**

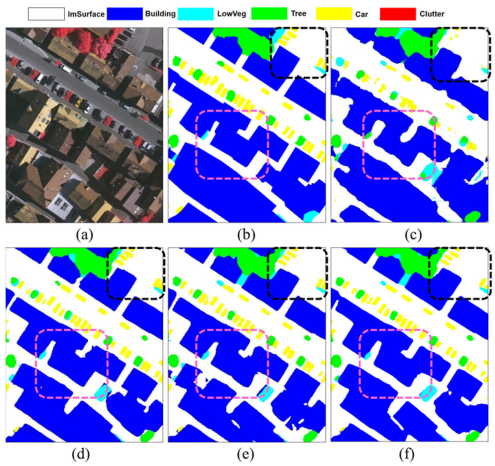


Fig. 9. Detailed visual comparison between GAMNet and other methods on Vaihingen tile 4. (a) Image. (b) Ground truth. (c) UZ_1. (d) CASIA2. (e) GSN3. (f) GAMNet.

1) SVL_3 [60]: It uses three bands together with NDSM, NDVI, and standard SVL-features to train the Adaboost-based classifier and then employs a CRF postprocessing step.
2) ADL_3 [61]: It combines a CNN trained with three bands, DSM, and NDSM and a random forest classifier trained with hand-crafted features, as well as a post-processing step.
3) UZ_1 [62]: Three bands and NDSM are used to train an encoder-decoder architecture with deconvolution as up-sampling method.
4) ONE_7 [63]: It uses three bands, NDVI, DSM, and NDSM to train two multiscale SegNets.
5) DLR_10 [42]: It uses three bands and DSM to achieve a multiscale ensemble learning of FCN, SegNet, and VGG.
6) CASIA2 [16]: Only three bands are used to train a self-cascaded network training and the ResNet101 in encoder part is pretrained on PASCAL VOC 2012 datasets.
7) GSN3 [53]: Only three bands are used to train a gated convolutional network and the ResNet101 in encoder part is pretrained on ImageNet.
8) DP-DCN [22]: Three bands and nDSM are used to train a DenseBlock based network.
9) TreeUNet [24]: It uses three bands and DSM to train a Tree-CNN block based network.
10) UFMG_4 [64]: Three bands and nDSM are used to train a dynamic multicontext network with dilated convolution.
11) PDM [65]: Three bands, DSM, and nDSM are used to train a network using DCNN predicted distance map.

In summary, only CASIA2 and GSN3 have the similar inputs of three spectral bands as our method, while more features, especially the DSM related features, are used by other methods.

The accuracies of the results from these models are given in Table IX. The proposed GAMNet achieves an *MF1* of 90.6% and an *OA* of 91.3% which surpasses the second highest method CASIA2 0.5% in *MF*1 and 0.2% in *OA*. We further compare the *F*1 for each category in detail, in which the *F*1 of car is improved the most by GAMNet. Compared with the second highest method of car, GAMNet achieves an *F*1 improvement of 1.8%. In order to visually compare the results produced by different methods, the segmentation results of tile 8, 6, 4, and 24 from the Vaihingen test set are selected and shown in Fig. 8. The methods of UZ_1, CASIA2, and GSN3 are selected for visual comparison because their accuracies cover a relatively wide range. Since the visual difference is not apparent in the whole image, we enlarge a part in tile 4 to present the difference clearly as in Fig. 9. As illustrated in the pink boxes, GAMNet has a better performance in localizing building boundaries. As illustrated in the black boxes, the cars are carefully separated by GAMNet and GSN3, but are sticking together by CASIA2 and are even missing by UZ_1. As a whole, the proposed GAMNet achieves a state-of-the-art performance on the Vaihingen dataset.

The comparison is then performed on the ISPRS Potsdam dataset. Only near infrared, red, and green bands are used for training the GAMNet network. The GAMNet network is also trained with patch size 512 × 512 pixels and the parameters λ of the composite loss is set to 0.8. The inference is performed by patches with 512 × 512 pixels with 75% overlay ratio which is the same with Vaihingen dataset. Five methods in Table IX are selected for comparison on the Potsdam test set, including SVL_3, UZ_1, CASIA2, TreeUNet, and UFMG_4.

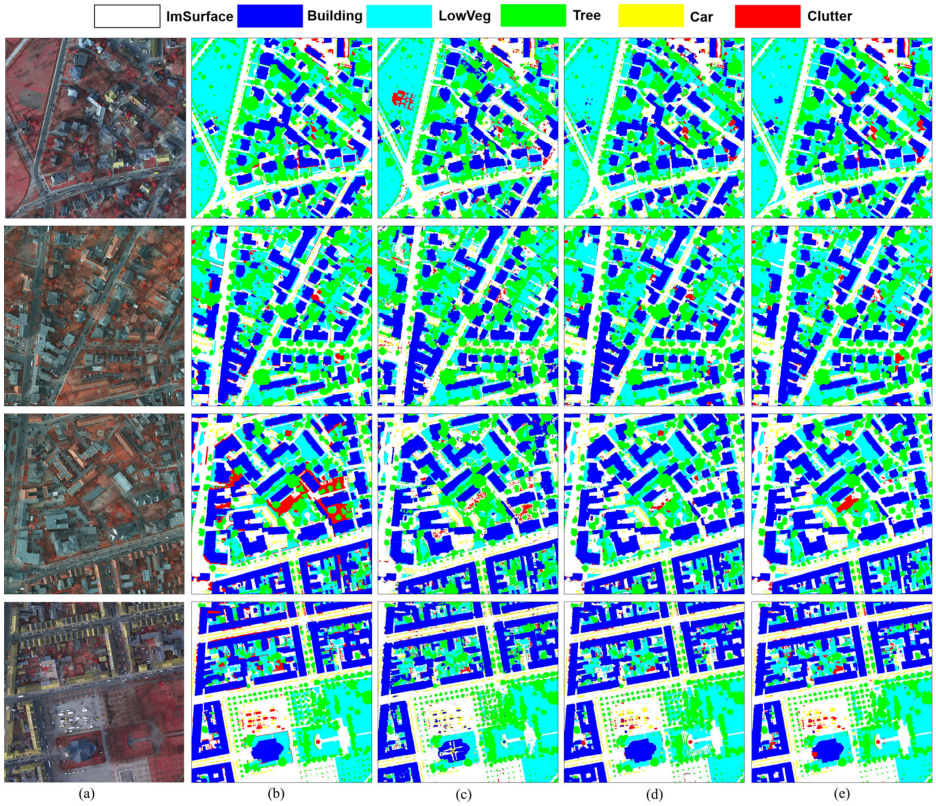


Fig. 10. Visual comparison between GAMNet and other methods on Potsdam tile 2_13, 3_13, 4_13, and 5_13 from test set. (a) Image. (b) Ground truth. (c) UZ_1. (d) CASIA2. (e) GAMNet.

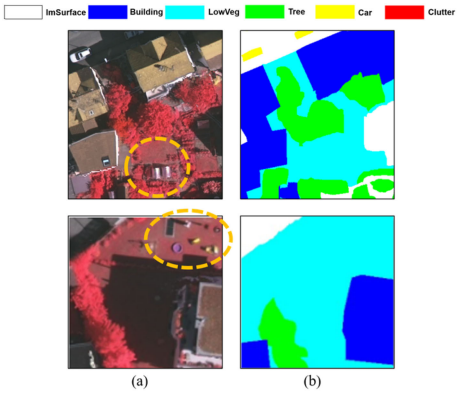


Fig. 11. Examples of mislabelled ground truth in the Vaihingen training set. (a) Image. (b) Ground truth.

We further add four methods of SWJ_2, HUSTW4, MCA [66], and ResUNet-a [25] for comparison. SWJ_2 and HUSTW4 are not published yet and only described in an abstract on the website of ISPRS. SWJ_2 uses only near infrared, red, and green bands to train a shortcut block based adaptive network with pretrained ResNet-101. HUSTW4 uses near infrared, red, green bands, DSM, and NDSM data to train a deconvolution network combined with U-Net. MCA uses four bands and DSM to train a multiscale context aggregation network. ResUNet-a uses four bands and nDSM to train a ResBlock based network.

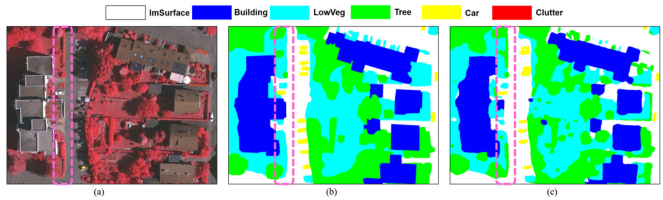


Fig. 12. Examples of misclassification between trees and low vegetation. (a) Image. (b) Ground truth. (c) GAMNet.

As given in Table X, GAMNet also achieves the highest accuracy indicated by *MF1*. GAMNet ranks the first place on the categories of car in *F*1. The segmentation results of tile 2_13, 3_13, 4_13, and 5_13 from the Potsdam test set are selected and shown in Fig. 10 for visual comparison, together with the results by UZ_1 and CASIA2.

V. Discussion

According to the experiments in Section IV, we demonstrated the effectiveness of the integration module, the composite loss function, and the patch size for both training and inferencing GAMNet. By comparing with other representative methods, it is proved that GAMNet can improve the accuracies of semantic segmentation of HR images with the benefit of combining the GM and AM, which simultaneously imposes boundary

TABLE XI
COMPARISON OF SEGMENTATION ACCURACIES BY INTEGRATING ASPP INTO GAMNET WITH DIFFERENT WAYS ON THE VAIHINGEN AND POTSDAM TEST SETS

Data	Model	*F*1 (%)					*MF*1 (%)	*MIoU* (%)	*OA* (%)
		ImSurface	Building	LowVeg	Tree	Car			
Vaihingen	GAMNet	93.02	95.62	84.73	90.45	87.55	90.28	82.50	91.15
	GAMNet+ASPP1	93.08	95.91	85.32	90.61	87.81	90.54	82.93	91.40
	GAMNet+ASPP2	92.89	95.64	84.96	90.59	87.71	90.36	82.63	91.19
Potsdam	GAMNet	93.63	97.43	88.25	89.01	96.57	92.98	87.11	91.52
	GAMNet+ASPP1	93.94	97.50	88.54	89.10	96.43	93.10	87.32	91.79
	GAMNet+ASPP2	93.93	97.47	88.47	89.06	96.53	93.09	87.30	91.75

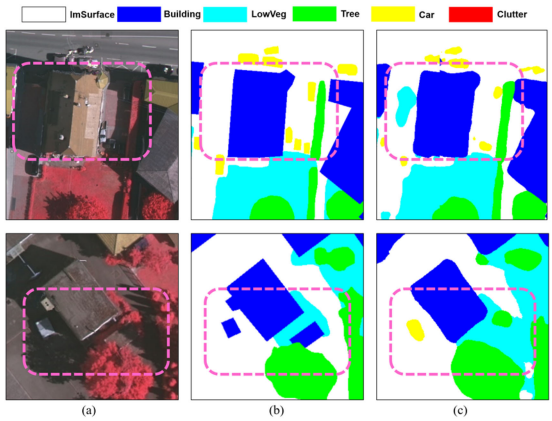


Fig. 13. Examples of misclassification in shaded areas. (a) Image. (b) Ground truth. (c) GAMNet.

constraint and improves the segmentation performance of small objects.

A. Typical Errors of GAMNet on ISPRS Dataset

Even though GAMNet achieves a state-of-the-art semantic segmentation performance on ISPRS datasets, it still has several typical errors that need to be further corrected.

1) *Label Errors*: Some pixels are mislabeled in the ground truth, which would lead to an underestimate of semantic segmentation accuracy. Two examples from Vaihingen dataset are shown in Fig. 11, where buildings marked by yellow circle are mislabeled as low vegetation in the upper row, and several objects that should be labeled as clutter are mislabeled as low vegetation in the bottom row.
2) Confusion between trees and low vegetation as shown in Fig. 12 could be due to the characteristic of high diversity in HR images. As marked by the pink boxes, it is hard to distinguish the true category purely through the spectral images with near infrared, red, and green bands. Most vegetation along the road is marked as low vegetation in the ground truth, but is segmented as trees. For this case, the inclusion of additional DSM related features would be beneficial.
3) Shaded areas are easily misclassified. By carefully observing the GAMNet results, we found that pixels in shaded areas are often misclassified. As illustrated in Fig. 13, where the pink boxes highlight the shaded areas, several cars in the shadow of buildings are misclassified as impervious surface. It also shows that a shaded building in the bottom row of Fig. 13 is classified as a car. The errors in shaded areas are very common in the Vaihingen dataset. Unfortunately, it still lacks an effective solution to this type of errors and remains to be explored in the future.

B. Visual Interpretation of GAM

To further understand the function and the effectiveness of the integration module, we visualize the feature maps in GM and AM. Two different types of examples with different characteristics are selected for visual interpretation of GAM. The example image in Fig. 14(a) has a large area of long and narrow vegetation, shadows, and buildings, representing the high diversity of geographic objects. Many small cars are scattered around the large buildings in Fig. 14(b), representing the various sizes of geographic objects.

Example weight maps obtained by the gate function are shown in Fig. 14. As the training epoch increases, the pixels with high gate weight tend to better match the object boundaries, which allows more information from multiscale feature maps to be used for improving boundary accuracy, and thus the network output appears to be closer to the ground truth. In addition, we can see that the weight maps at different levels present different details. From the highest level GAM1 to the lowest level GAM3, more boundary details are presented in the weight map. This is because the low-level feature map itself retains more details. Moreover, as is shown in Fig. 14(b), small cars in high gate weight maps can catch precise boundaries, which demonstrates the effectiveness of GM.

Example attention maps obtained by the AM are shown in Fig. 15. As the training epoch increases, the pixels with high attention values tend to cluster in the same category, which can be viewed as an optimization process. In addition, the convolutional features at different levels tend to have attention on different objects, which is consistent with the explanation in Section II-B.

C. Joinability of Multiscale Structures

The skip-net is adopted as the basic multiscale network for GAMNet to generate multiscale features. Here, we further study the joinability of skip-net and pyramid pooling net since the skip-net fuses all the intermediate feature levels while the pyramid pooling net only processes the high-level features.

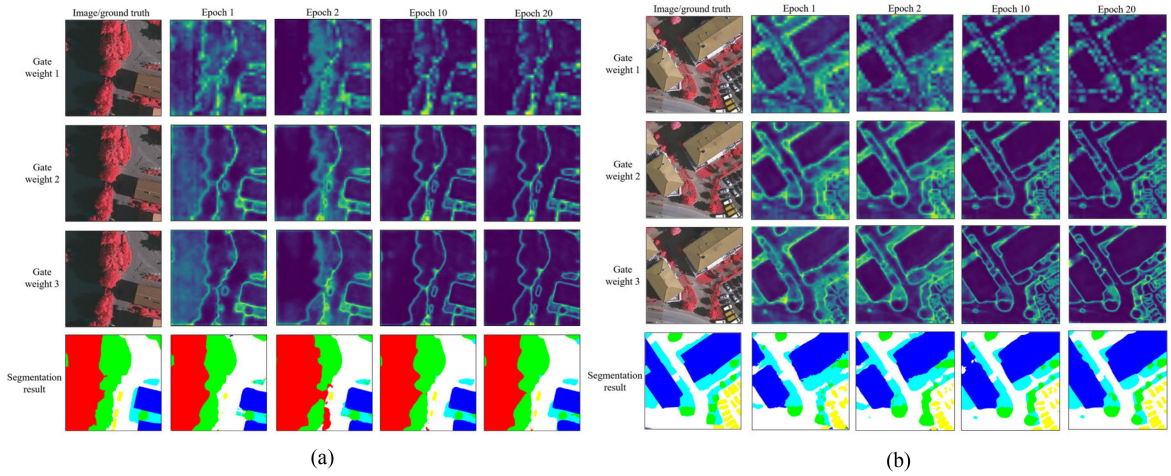


Fig. 14. Weight maps produced by the gate function, where weight 1, 2, and 3 correspond to different feature levels from high to low. The color in the weight map indicates the gate weight value, and a lighter color indicates a higher value. Epoch refers to the training iteration and the segmentation result is the output of the network in the corresponding training epoch.

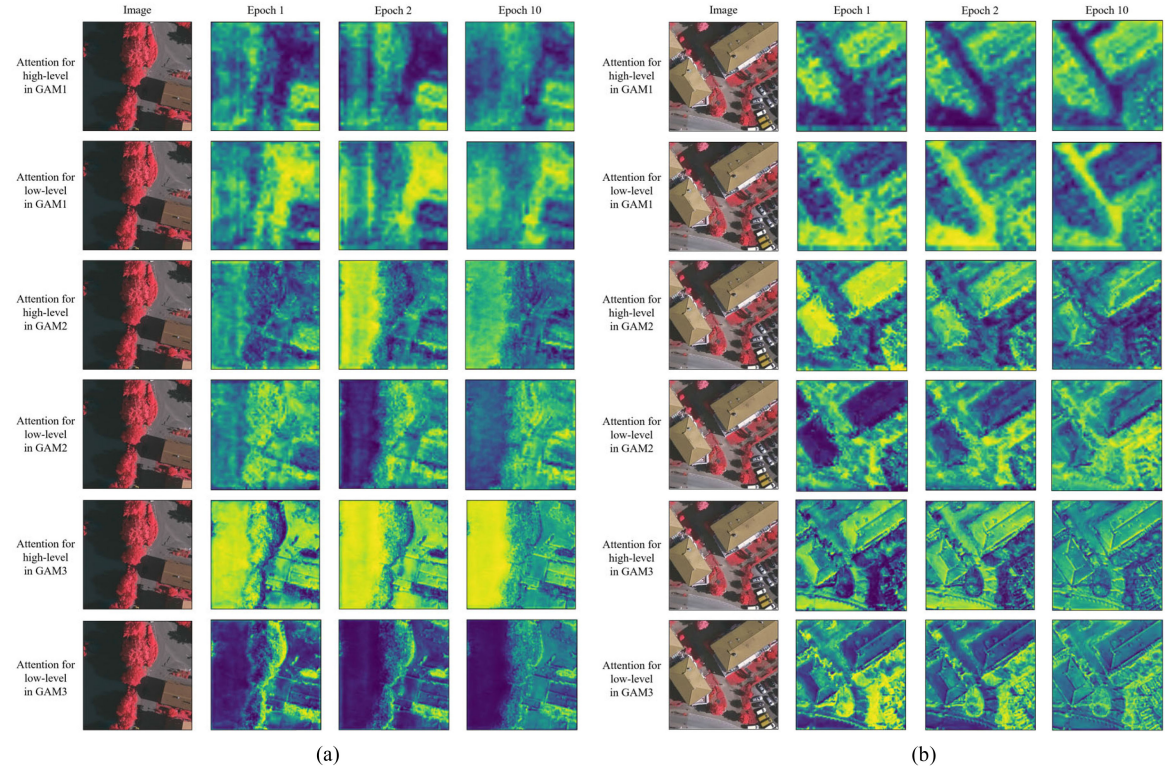


Fig. 15. Visualization of attention at different feature levels, where the attention for high-level and for low-level comes from the feature map in the encoder and the decoder, respectively. GAM 1, 2, and 3 refer to the integration module at different feature levels from high to low. The color from blue to yellow in the attention map indicates the attention value changing from low to high.

Specifically, we introduce ASPP, which is an effective pyramid pooling net, into GAMNet. ASPP sets up parallel dilated convolution with different dilated rates. We set the dilated rate series as {6, 12, 18}. In order to insert ASPP module into GAMNet, two different architectures are designed, as shown in Fig. 16. The main difference between them is whether the ASPP module is taken as a separate scale in the encoder part. Fig. 16(a) employs the ASPP module to deeply exploit the multiscale features from high-level features. We name it as GAMNet with ASPP1. GAMNet with ASPP2 takes ASPP as a separate scale after the four blocks of ResNet-101, which is similar to the network proposed in [55]. The parameter λ of the composite loss is set as 0.5. The segmentation accuracies of these modules with the MG, DA, and OI with 75% overlay ratio strategies on both the Vaihingen and Potsdam test sets are given in Table XI. The design of both the ASPP1 and ASPP2 is

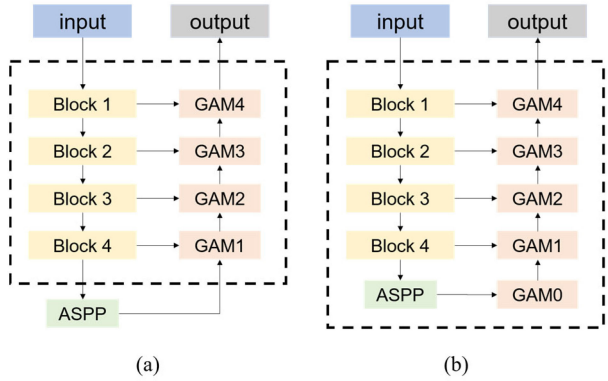


Fig. 16. Two ways of integrating ASPP into GAMNet. (a) GAMNet with ASPP1. (b) GAMNet with ASPP2.

beneficial to the segmentation performance. For Vaihingen test set, GAMNet with ASPP1 reveals the best performance among the three modules. Especially, it mainly improves the accuracy of low vegetation by 0.59% in $F1$. For the Potsdam test set, GAMNet with ASPP1 and ASPP2 have similar performance, which outperform GAMNet by 0.1% in *MF1* and 0.2% in *MIoU*. This is because the high-level features after block 4 of ResNet-101 and the multiscale features after ASPP form a scale gradient. Accordingly, the design of GAMNet with ASPP can be seen as an effective combination of skip-net and pyramid pooling net, which holds the potential of achieving higher segmentation accuracies by the hybrid skip-net and pyramid pooling net.

VI. Conclusion

In this article, we proposed a novel end-to-end network for semantic segmentation of HR images by designing and inserting a plug-and-play integration module GAM in a skip-net. For the GM in GAM, the information entropy is taken as the gate function to impose constraints on object boundaries. The AM in GAM is employed to select effective features from different levels and thus to enhance the weight of useful features. Hence, the integration module GAM is able to simultaneously impose boundary constraint and improve segmentation performance for small objects. A composite loss function is specially designed by combining the cross entropy loss and the loss from the GM, which is beneficial to the model optimization and improves the performance of the integration module.

We evaluate the proposed GAMNet on the ISPRS 2-D semantic datasets. Extensive experiments prove that GAMNet achieves state-of-the-art performance compared with other published results. According to both the quantitative and qualitative evaluation results, we can conclude that GAMNet can produce segmentation results with accurate boundaries and apparently improves the segmentation accuracy for small objects. Specifically, the segmentation accuracy of cars is apparently improved by GAMNet, which is generally considered as more difficult to segment than other categories.

In the future, we consider to expand the input data, such as NDSM, to overcome the confusing problem. Further, we will pay more attention to the serious misclassification in shaded areas which still has no good solution so far.

References

[1] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017, doi: 10.1109/MGRS.2017.2762307.

[2] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state-of-the-art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016, doi: 10.1109/MGRS.2016.2540798.

[3] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019.

[4] B. Xi, J. Li, Y. Li, R. Song, W. Sun, and Q. Du, "Multiscale context-aware ensemble deep KELM for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3022029.

[5] L. Wang, J. Peng, and W. Sun, "Spatial–Spectral Squeeze-and-Excitation residual network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 7, Apr. 2019, Art. no. 884.

[6] M. Wang, H. Zhang, W. Sun, S. Li, F. Wang, and G. Yang, "A Coarse-to-Fine deep learning based land use change detection method for high-resolution remote sensing images," *Remote Sens.*, vol. 12, no. 12, Jun. 2020, Art. no. 1933.

[7] Q. Geng and Z. Zhou, "Survey on recent progresses of semantic image segmentation with CNNs," in *Proc. Int. Conf. Virtual Reality Visual.*, Sep. 2016, pp. 158–163.

[8] J. Xin, X. Zhang, Z. Zhang, and W. Fang, "Road extraction of high-resolution remote sensing images derived from denseunet," *Remote Sens.*, vol. 11, no. 21, Oct. 2019, Art. no. 2499.

[9] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded End-to-End convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017, doi: 10.1109/TGRS.2017.2669341.

[10] Y. Yi, Z. Zhang, W. Zhang, C. Zhang, W. Li, and T. Zhao, "Semantic segmentation of urban buildings from VHR remote sensing imagery using a deep convolutional neural network," *Remote Sens.*, vol. 11, no. 15, Jul. 2019, Art. no. 1774.

[11] Y. Shi, Q. Li, and X. X. Zhu, "Building segmentation through a gated graph convolutional neural network with deep structured feature embedding," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 184–197, Jan. 2020.

[12] Y. Chen, Y. Li, J. Wang, W. Chen, and X. Zhang, "Remote sensing image ship detection under complex sea conditions based on deep semantic segmentation," *Remote Sens.*, vol. 12, no. 4, Feb. 2020, Art. no. 625.

[13] M. Wieland, Y. Li, and S. Martinis, "Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network," *Remote Sens. Environ.*, vol. 230, Sep. 2019, Art. no. 111203.

[14] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang, "Cloud and cloud shadow detection in landsat imagery based on deep convolutional neural networks," *Remote Sens. Environ.*, vol. 225, pp. 307–316, May 2019.

[15] L. Weng, Y. Xu, M. Xia, Y. Zhang, J. Liu, and Y. Xu, "Water areas segmentation from remote sensing images using a separable residual segnet network," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, Apr. 2020, Art. no. 256.

[16] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan, "Semantic labeling in very high resolution images via a self-cascaded convolutional neural network," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 78–95, Nov. 2018.

[17] Q. Yao, X. Hu, and H. Lei, "Geospatial object detection in remote sensing images based on multi-scale convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 1450–1453.

[18] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "High-Resolution aerial image labeling with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7092–7103, Dec. 2017, doi: 10.1109/TGRS.2017.2740362.

[19] Q. Liu, M. Kampffmeyer, R. Jenssen, and A.-B. Salberg, "Dense dilated convolutions' merging network for land cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6309–6320, Sep. 2020, doi: 10.1109/TGRS.2020.2976658.

[20] N. Audebert, B. L.. Saux, and S. Lefèvre, "Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 140, pp. 20–32, Jun. 2018.

[21] Z. Cao *et al.*, "End-to-End DSM fusion networks for semantic segmentation in high-resolution aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1766–1770, Nov. 2019, doi: 10.1109/LGRS.2019.2907009.

[22] C. Peng, Y. Li, L. Jiao, Y. Chen, and R. Shang, "Densely based multi-scale and multi-modal fully convolutional networks for high-resolution

remote-sensing image semantic segmentation," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2612–2626, Aug. 2019, doi: 10.1109/JSTARS.2019.2906387.

[23] M. Zhang, X. Hu, L. Zhao, Y. Lv, M. Luo, and S. Pang, "Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images," *Remote Sens.*, vol. 9, no. 5, May 2017, Art. no. 500.

[24] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "TreeUNet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 1–13, Oct. 2019.

[25] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogramm. Remote Sens.*, vol. 162, pp. 94–114, Apr. 2020.

[26] G. Chen, X. Zhang, Q. Wang, F. Dai, Y. Gong, and K. Zhu, "Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of very-high-resolution remote sensing images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 5, pp. 1633–1644, May 2018, doi: 10.1109/JSTARS.2018.2810320.

[27] X. Zheng, L. Huan, G.-S. Xia, and J. Gong, "Parsing very high resolution urban scene images by learning deep convnets with edge-aware loss," *ISPRS J. Photogramm. Remote Sens.*, vol. 170, pp. 15–28, Dec. 2020.

[28] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3640–3649.

[29] S. Yang and G. Peng, "Attention to refine through multi scales for semantic segmentation," in *Proc. Adv. Multimedia Inf. Process.*, 2018, pp. 232–241.

[30] H. Lin, Z. Shi, and Z. Zou, "Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network," *Remote Sens.*, vol. 9, no. 5, May 2017, Art. no. 480.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," Apr. 2017, Accessed: Dec. 14, 2019. [Online]. Available: http://arxiv.org/abs/1612.01105

[32] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," Jun. 2017, Accessed: Sep. 28, 2019. [Online]. Available: http://arxiv.org/abs/1706.05587

[33] W. Kang, Y. Xiang, F. Wang, and H. You, "EU-Net: An efficient fully convolutional network for building extraction from optical remote sensing images," *Remote Sens.*, vol. 11, no. 23, Nov. 2019, Art. no. 2813.

[34] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," May 2015, Accessed: Oct. 13, 2019. [Online]. Available: http://arxiv.org/abs/1505.07293

[35] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," Nov. 2016, Accessed: Oct. 14, 2019. [Online]. Available: http://arxiv.org/abs/1611.06612

[36] Y. Liu, D. M. Nguyen, N. Deligiannis, W. Ding, and A. Munteanu, "Hourglass-ShapeNetwork based semantic segmentation for high resolution aerial imagery," *Remote Sens.*, vol. 9, no. 6, May 2017, Art. no. 522.

[37] F. Yu and V. Koltun, "Multi-Scale context aggregation by dilated convolutions," Nov. 2015, Accessed: Oct. 12, 2019. Online]. Available: http://arxiv.org/abs/1511.07122

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," May 2015, Accessed: Sep. 16, 2019. [Online]. Available: http://arxiv.org/abs/1505.04597

[39] N. Audebert, B. L.. Saux, and S. Lefevre, "How useful is region-based classification of remote sensing images in a deep learning framework?," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 5091–5094.

[40] Y. Fu *et al.*, "Mapping impervious surfaces in town–rural transition belts using China's GF-2 imagery and object-based deep CNNs," *Remote Sens.*, vol. 11, no. 3, Jan. 2019, Art. no. 280.

[41] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-Aware feature propagation for scene segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6818–6828.

[42] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015, Accessed: Oct. 21, 2019. [Online]. Available: http://arxiv.org/abs/1512.03385

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[45] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with atrous separable convolution for semantic image segmentation," Feb. 2018, Accessed: Sep. 28, 2019. [Online]. Available: http://arxiv.org/abs/1802.02611

[46] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3156–3164.

[47] P. Zhou *et al.*, "Attention-Based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 207–212.

[48] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4476–4484.

[49] Z. Liu, P. Gong, and J. Wang, "Attention-Based feature pyramid network for object detection," in *Proc. 8th Int. Conf. Comput. Pattern Recognit.*, Oct. 2019, pp. 117–121.

[50] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 603–612.

[51] K. Xu *et al.*, "Show, attend and tell: Neural image captiongeneration with visual attention," in *Proc. IEEE Int. Conf. Machine Learning (ICML)*, 2015, pp. 2048–2057.

[52] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Trans. Multimed.*, vol. 19, no. 6, pp. 1245–1256, Jun. 2017, doi: 10.1109/TMM.2017.2648498.

[53] H. Wang, Y. Wang, Q. Zhang, S. Xiang, and C. Pan, "Gated convolutional neural network for semantic segmentation in high-resolution images," *Remote Sens.*, vol. 9, no. 5, May 2017, Art. no. 446.

[54] M. A. Islam, M. Rochan, S. Naha, N. D. B. Bruce, and Y. Wang, "Gated feedback refinement network for Coarse-to-Fine dense semantic image labeling," Jun. 2018, Accessed: Dec. 14, 2019. [Online]. Available: http://arxiv.org/abs/1806.11266

[55] X. Li, H. Zhao, L. Han, Y. Tong, S. Tan, and K. Yang, "Gated fully fusion for semantic segmentation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 11418–11425, Apr. 2020, doi: 10.1609/aaai.v34i07.6805.

[56] T. H. Trinh, A. M. Dai, M.-T. Luong, and Q. V. Le, "Learning longer-term dependencies in RNNS with auxiliary losses," in *Proc. 35th Int. Conf. on Machine Learning*, 2018, pp. 4965–4974.

[57] M. Cramer, "The DGPF-Test on digital airborne camera evaluation – overview and test design," *Photogramm. - Fernerkund. – Geoinf.*, vol. 2010, no. 2, pp. 73–82, May 2010.

[58] F. Rottensteiner *et al.*, "The ISPRS benchmark on urban object classification and 3D building reconstruction," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 1–3, pp. 293–298, Jul. 2012.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. F. Li, ImageNet: A large-scale hierarchical image database, 2009.

[60] M. Gerke, "Use of the stair vision library within the ISPRS 2D semantic labeling benchmark (Vaihingen)," 2015.

[61] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. van den Hengel, "Semantic labeling of aerial and satellite imagery," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2868–2881, Jul. 2016, doi: 10.1109/JSTARS.2016.2582921.

[62] M. Volpi and D. Tuia, "Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017, doi: 10.1109/TGRS.2016.2616585.

[63] N. Audebert, B. L. Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and Multi-scale deep networks," Sep. 2016, Accessed: Nov. 4, 2020. [Online]. Available: http://arxiv.org/abs/1609.06846

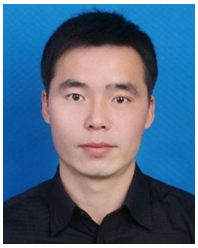
[64] K. Nogueira, M. Dalla Mura, J. Chanussot, W. R. Schwartz, and J. A. dos Santos, "Dynamic multicontext segmentation of remote sensing images based on convolutional networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7503–7520, Oct. 2019, doi: 10.1109/TGRS.2019.2913861.

[65] D. Chai, S. Newsam, and J. Huang, "Aerial image semantic segmentation using DCNN predicted distance maps," *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 309–322, Mar. 2020.

[66] J. Zhang, S. Lin, L. Ding, and L. Bruzzone, "Multi-Scale context aggregation for semantic segmentation of remote sensing images," *Remote Sens.*, vol. 12, no. 4, Feb. 2020, Art. no. 701.

Zixian Zheng received the B.S. degree in geographic information science from Sun Yat-sen University, Guangzhou, China, in 2019. She is currently working toward the M.S. degree in cartography and geographical information system from Nanjing University, Nanjing, China.

Her research interests include semantic segmentation and deep learning for remote sensing.



Xueliang Zhang (Member, IEEE) received the B.S. degree in geographical information system and the Ph.D. degree in remote sensing of resources and environment from Nanjing University, Nanjing, China, in 2010 and 2015, respectively.

From 2014 to 2015, he was a Visited Student with Informatics Institute, University of Missouri, Columbia, MO, USA. From 2016 to 2018, he was an Associate Researcher with the Department of Geographic Information Science, Nanjing University. He is currently an Associate Professor with the Department of Geographic Information Science, Nanjing University. His research interests include high-resolution remote sensing image analysis, semantic segmentation, and deep learning for remote sensing.

Pengfeng Xiao (Senior Member, IEEE) received the B.M. degree in land resource management from Hunan Normal University, Changsha, China, in 2002, and the Ph.D. degree in cartography and geographical information system from Nanjing University, Nanjing, China, in 2007.

From 2007 to 2009, he was a Lecturer with the School of Geography and Ocean Science at Nanjing University, where he was an Associate Professor, from 2010 to 2018. Since 2019, he has been a Professor with Nanjing University. He was a Visiting Scholar with the Department of Geography, University of Giessen, Giessen, Hesse, Germany, from 2011 to 2012, and the Department of Environmental Science, Policy, and Management, University of California at Berkeley, Berkeley, CA, USA, from 2014 to 2015. He has authored or coauthored four books and more than 60 articles. His current research interests include high-resolution remote sensing image analysis, remote sensing of snow cover, and land use and land cover change.

Zhenshi Li received the B.S. degree in geographic information science from Hohai University, Nanjing, China, in 2019. He is currently working toward the M.S. degree in cartography and geographical information system from Nanjing University, Nanjing, China.

His research interests include semantic segmentation and weakly supervised deep learning for remote sensing.