Hindawi
Journal of Sensors
Volume 2021, Article ID 6659831, 14 pages
https://doi.org/10.1155/2021/6659831

Research Article

Remote Sensing Image Scene Classification Based on Fusion Method

Liancheng Yin,[1] Peiyi Yang,[2] Keming Mao,[1] and Qian Liu[1]

[1]*College of Software, Northeastern University, Shenyang, Liaoning Province 110004, China*
[2]*College of Computer Science, University of Virginia, Charlottesville, Virginia 22904, USA*

Correspondence should be addressed to Keming Mao; maokm@swc.neu.edu.cn

Received 20 October 2020; Revised 12 January 2021; Accepted 15 January 2021; Published 9 June 2021

Academic Editor: Zhenxing Zhang

Remote sensing image scene classification is a hot research area for its wide applications. More recently, fusion-based methods attract much attention since they are considered to be an useful way for scene feature representation. This paper explores the fusion-based method for remote sensing image scene classification from another viewpoint. First, it is categorized as front side fusion mode, middle side fusion mode, and back side fusion mode. For each fusion mode, the related methods are introduced and described. Then, classification performances of the single side fusion mode and hybrid side fusion mode (combinations of single side fusion) are evaluated. Comprehensive experiments on UC Merced, WHU-RS19, and NWPU-RESISC45 datasets give the comparison result among various fusion methods. The performance comparisons of various modes, and interactions among different fusion modes are also discussed. It is concluded that (1) fusion is an effective way to improve model performance, (2) back side fusion is the most powerful fusion mode, and (3) method with random crop+multiple backbone+average achieves the best performance.

1. Introduction

With explosive increasing of remote sensing data, analysis and processing remote sensing image effectively and efficiently becomes of great importance. Remote sensing image scene classification, which aims to classify remote sensing image into different types based on image content, has been attracted more and more attentions for its comprehensive application in fields of geography, ecology, city plan, forest monitor, military, etc [1].

Remote sensing image scene classification essentially belongs to domains of machine learning and computer vision. With well-organized training dataset, models can be learned through minimizing loss functions between model output and ground-truth label. According to feature extraction and representation techniques, existing methods can be categorized into three types: method based on low-level feature, method based on mid-level feature, and method based on deep feature.

Methods based on low-level feature focus on image color, texture, shape, intensity, or their combinations, which are intuitive representations of scene image. Color histogram was used to extract global feature in [2, 3]. In [4], Gabor filters with various scales and orientations were used to represent image texture. In [5], global morphological texture features were extracted with circular covariance histogram, rotation-invariant point triplets, and their extensions with Fourier power spectrum. The superpixel-based Extended Random Walker spectral–spatial classification method was proposed in [6]. It did not only incorporate superpixel to cluster local similar pixels but also build up relationship between superpixels.

Mid-level feature, such as local binary pattern (LBP), scale-invariant feature transform (SIFT), histogram of oriented gradients (HOG), and spatial envelop (Gist), was well engineered by experienced experts [7–13]. In [7], multiscale LBP features were extracted from dense patches. Fisher vector encoding and extreme learning machine were then used for the training classification model. A multineighborhood LBP method was proposed for feature extraction in small image patches [8]. Bag of visual words (BoVW) and support vector machine (SVM) were used as feature representation

and classifier. SIFT was adopted in [9] for classification of remote sensing imagery, and it showed good performance. In [10], a generalized principal component analysis (PCA) was used to reduce dimension of SIFT, and more spatial locality information can be preserved. Ship-HOG was proposed to describe target in scene image by aligning the axis to the vertical direction [11]. In [12], the rotation invariant HOG-based method was proposed by constraining the similarity between training samples with various rotations. In [13], candidate regions were obtained with segmentation. Then, gist features were extracted, and random forest classifier was trained.

Low-level feature and mid-level feature are known as traditional methods. They extract and represent features of remote sensing image in the shallow layer, and the complex structure cannot be captured. On the other hand, they treat feature extraction and classifier construction as two separate stages, which make the model suboptimal. With the breakthrough of machine learning technology [14] and massive training data (ImageNet, COCO dataset [15, 16]), deep learning models achieve great success in computer vision, including AlexNet [17], ZFNet [18], VGGNet [19], InceptionNet [20], and ResNet [21]. Deep neural network model-based methods become prevalent in field of remote sensing image scene classification. Sparse autoendocer models combined with convolutional features were proposed in [22, 23]. In [24], pairwise constraints were integrated into a stacked autoencoder, and more informative features can be extracted. The traditional BoVW method was improved in [25]. User-defined features were replaced with deep features which were extracted with an off-the-shelf CNN model. In [26], a CNN model was trained by minimizing entropy loss along with a metric learning regularization, which enforced the model more discriminative. Salient regions were extracted with visual attention, and these regions were used to learn initial parameters of the CNN model [27] which was subsequently fine tuned. Attention mechanism was also introduced in [28], which can discard unimportant information. The frequency domain was incorporated in [29], and representations of space and frequency were combined for scene classification. A knowledge distillation-based method was proposed to train a powerful light CNN model for scene classification [30]. Siamese-GAN was proposed for aerial vehicle image classification with crossdomain conditions [31]. GAN methods were also adopted in [32, 33]. Large scale of synthetic remote sensing image samples was generated, and dataset can be expanded with GAN. Comparing with traditional machine learning methods, the advantage of deep learning-based scene image classification is mainly owing to the fact that it can extract more complex and appropriate feature combination structures. Then, important and discriminate feature representation can be obtained with deep layers. and the irrelevant variants are ignored. More works about remote sensing image scene classification with deep learning can be referred to [34, 35].

Among these deep learning models, fusion-based methods are widely used and are gaining more popularity. A multiscale deep feature method was proposed in [36]. First, input satellite images were wrapped into multiple scales, and each one was represented with a deep feature through pretrained model. Then, SVM classifier was trained with using multikernel learning. Two separate nets were trained with labeled images and their rescaled samples [37]. These two models shared the same backbone network, and the similarity between them was added to loss function as a regularization term. A two-stage deep feature fusion method was proposed in [38], which focused on feature combination from various layers of deep networks. In [39], three CNN models were trained with different receptive fields. The final classification result was determined with maximum posterior probability of the outputs of three independent models. The DropBand technique was proposed in [40]. Training images were generated by dropping certain spectral bands. Images with the same spectral band set were used to train a separate CNN model. The final classification result was obtained by averaging all model outputs. A four-layer feature representation was constructed with two convolutional layers and two full connected layers of a VGGNet model [41]. Then, canonical correlation analysis (CCA) was used for feature fusion. In [42], encoded mixed-resolution representation was proposed by concatenation of low-level, mid-level, and fully connected feature. They were further encoded by locally aggregated descriptors and average pooling. In [43], each layer of the VGGNet model was extracted as separated feature descriptors. Then, they were combined to construct the final representation with CCA fusion. In [44], a feature encoding module was designed to aggregate the intermediate convolutional features into the final representation. In [45], deconvolution module was adopted to learn parameter weight in an unsupervised way, and the spatial pyramid model was used for feature aggregation. A bidirectional adaptive feature fusion strategy was investigated in [46]. Embedded deep feature and SIFT feature were fused with a recurrent neural network structure. A gradient boosting random CNN framework was adopted for efficient training multiple deep models [47]. The hydra model was proposed in [48]. Hydra's body was first designed to providing a good starting point for optimization, and then hydra's heads were trained to from the ensemble of CNNs.

Although the concept "fusion" is covered in all these works, different ways of expressions are given, i.e., "multilayer features," "feature fusion," "multi-scale deep feature," "multilevel fusion," "multi-stage feature fusion," "feature combination," "ensemble of model," and "feature aggregation." As far as we know, there is no unified and standard description, and this imprecise expression is inconvenient for further research. To address this problem, this paper makes a study on fusion-based remote sensing image scene classification in another view. According to the source of fusion (input data, feature map or vector, model output, etc.), three types of fusion modes, front side fusion mode, middle side fusion mode, and back side fusion mode, are given. In addition, typical fusion methods of three fusion modes are described. Experimental evaluations are conducted based on three widely used datasets. Comprehensive comparisons of model accuracy and efficiency are given among different fusion methods. Relationship and influence among different fusion mode are further analyzed. The main contributions of this research are twofold:

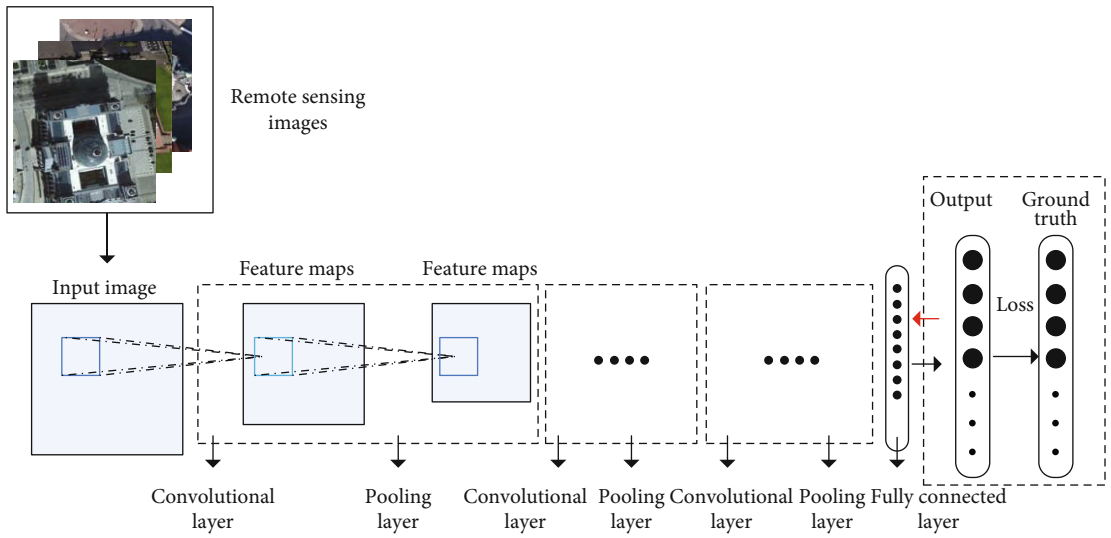

Figure 1: Typical structure of remote sensing image scene classification based on the CNN model. Blank boxes represent similar operations in CNNs.

(i) This work makes study on remote sensing image sense classification based on the origin of fusion operation. Representative works are summarized, and three types of fusion mode are defined

(ii) Extensive experimental evaluation and analysis are carried out. Importance of various fusion mode is shown quantitatively

The rest of this paper is organized as follows. Section 2 describes the workflow of remote sensing image scene classification based on the CNN model. Three types of fusion mode and their typical methods are given in Section 3. Section 4 demonstrates experimental evaluation and analysis. Section 5 concludes this paper.

2. Workflow of Remote Sensing Image Scene Classification Based on CNNs

This paper makes study on remote sensing image scene classification based on fusion methods. We first give the basic workflow based on the CNN model in this section.

Figure 1 demonstrates a typical workflow of remote sensing image scene classification based on the CNN model. The basic component of the CNN model includes convolutional layer, pooling layer, and fully connected layer. The input training image samples are fed into the CNN model, then after a series of convolutional and pooling operations, the final output is obtained with a fully connected layer using the softmax function. The loss function between model output and ground truth label is used for tuning the weight of the model parameter. The training process will be stopped when the loss value is less than a prespecified threshold. This structure can also be seen as a fundamental workflow of feature extraction and representation using the CNN model on general image classification problem. For its excellent performance, the CNN based model has become a dominant method. In this study, we only consider fusion methods that based on CNN models.

3. Fusion-Based Remote Sensing Image Scene Classification

Fusion usually incorporates information from different modalities, and they are combined for further processing. In our study, CNN models incorporated with various fusion methods are treated as a multistage procedure. These fusion methods are further divided into three types according to the stage where fusion operation works.

3.1. Remote Sensing Image Scene Classification Based on the Front Side Fusion Mode. For remote sensing image scene classification, the scale, position, and contextual information of specific object change greatly between image samples. To handle this problem, some methods focus on the input image data. It is also called multiscale, multilevel, image fusion, data fusion, etc. It is essentially a process of integrating multiple image sources to produce more consistent, accurate, and useful information than that provided by individual one. It is expected that the fused data is more informative and synthetic. In this paper, front side fusion mode is used to describe this type of fusion for the training image samples that are processed before they are fed into CNN models.

Figure 2 demonstrates remote sensing image scene classification methods based on the front side fusion mode. As is shown in Figure 2(a), original images are randomly cropped, and the patches are used as training data. This method was adopted in [37, 40]. Original images are cropped with multiple scales, and then they are resized with the same dimensions. This is shown in Figure 2(b), and it was used in [36, 39]. Figure 2(c) gives another example [40]. Original images are decomposed into many channels, such as RGB channel, HSV channel, and YCbCr channel, as well as gray and binary

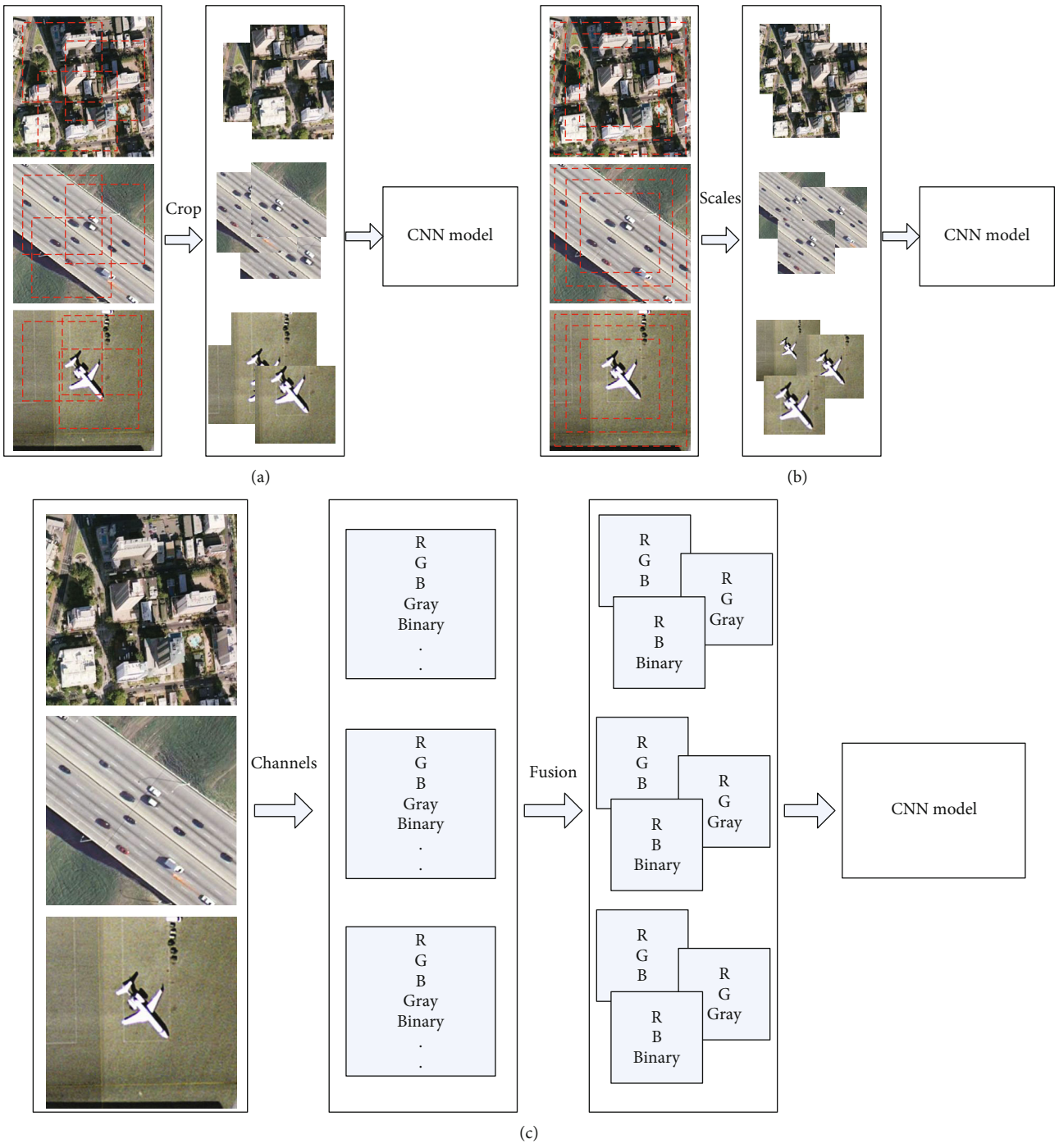

Figure 2: Demonstrations of three methods for remote sensing scene image classification based on front side fusion mode. (a) Front side fusion with random crop of input image. (b) Front side fusion with multiple scales of input image. (c) Front side fusion with channel combination of input image.

formats. Groups of channel subsets are selected to form the new training samples.

3.2. Remote Sensing Image Scene Classification Based on the Middle Side Fusion Mode. Middle side fusion can be regarded as feature level fusion. It gathers features from different paths or branches of CNN network and combines them together into a single feature map or vector. The fused feature is either fed into a classifier or used for further transformation. The difficulties for middle side fusion are the integration of heterogeneous feature.

Figure 3 demonstrates methods for remote sensing image scene classification based on middle side fusion. As is shown in Figure 3(a), C_n means nth convolutional layer, and P_n means nth pooling layer. f_n means feature map after operations of C_n and P_n. f_{n-1} and f_{n+1} are feature maps from the

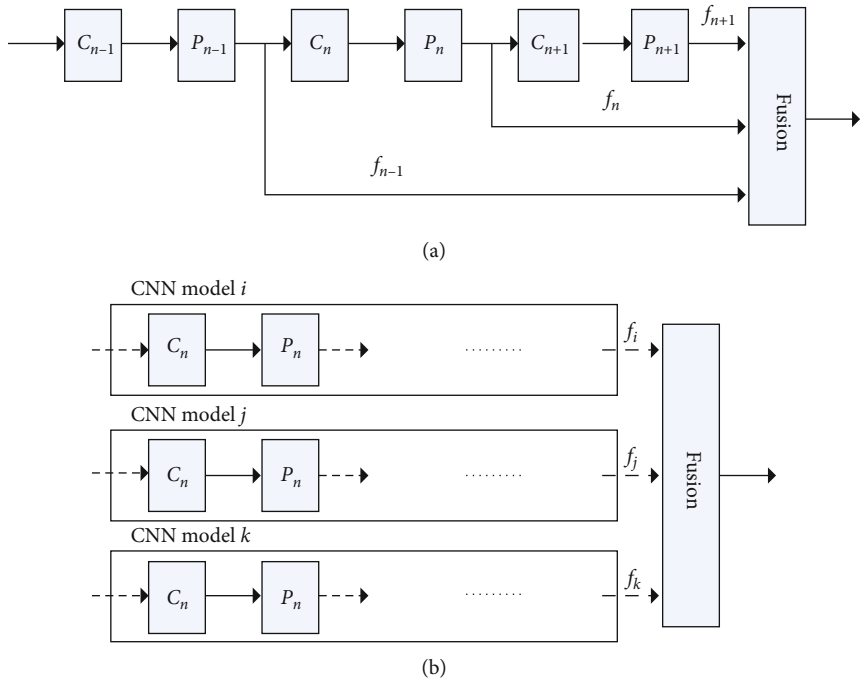

Figure 3: Demonstrations of remote sensing image scene classification based on the middle side fusion mode. (a) Middle side fusion with the single backbone network. (b) Middle side fusion with multiple backbone networks.

former and latter layers. A fused feature can be constructed by combining f_{n-1}, f_n, and f_{n+1}. This method is adopted in [38, 41]. An alternative way is shown in Figure 3(b). Feature maps f_i, f_j, and f_k are final outputs or intermediate outputs from the CNN model i, CNN model j, and CNN model k, respectively. Then, the fused feature is constructed by combining f_i, f_j, and f_k, which are from various paths. This method is adopted in [36–38, 48].

$$f_{\text{fusion}} = w_i \times f_i + w_j \times f_j + w_k \times f_k. \tag{1}$$

Equation (1) shows the linear combination of feature fusion computation. Here, feature maps f_i, f_j, and f_k are set with the same dimension. They are linearly combined at the same index, and f_{fusion} has the same dimension. w_i, w_j, and w_k are corresponding weights.

$$f_{\text{fusion}} = \text{concat}\left[f_i, f_j, f_k\right]. \tag{2}$$

Equation (2) gives another feature fusion computation and concatenation. Features are stacked instead of linear computation. f_{fusion} is the concatenation result of f_i, f_j, and f_k. If f_i, f_j, and f_k are set with dimensions of $10 \times 10 \times 3$, then f_{fusion} is a feature map set with $10 \times 10 \times 9$.

Multiple kernel learning (MKL) was also used for feature fusion computation [49]. The goal is to learn the fusion weights for each feature automatically. Let $x^i = \{x_1^i, \cdots, x_j^i, \cdots, x_n^i\}$ denote feature of an image sample i, which is composed of multiple feature x_j^i. The MKL method generates new feature vector $\{d_1 x_1^i, \cdots, d_j x_j^i, \cdots, d_n x_n^i\}$ that is optimal for training SVM classifier. It can be obtained by minimizing parameters $1/2\ \|w\|$ under constraints of $y^i(<\phi(x^i), w>+b) \geq 1$ for all training samples $\{x^i, y^i\}$. ϕ is a nonlinear mapping function. The key problem is to solve $<\phi(x^i), \phi(x^j)>$. However, it is hard to compute ϕ directly. While using kernel trick, it can be computed as follows.

$$<x^i, y^i> = K\left(x^i, x^j\right) = \sum_{m=1}^{N} d_m^2 K_m\left(x_m^i, x_m^j\right). \tag{3}$$

As shown in equation (3), $K()$ is the kernel function. Based on this scheme, d_m^2 can be computed by the gradient descend method.

3.3. Remote Sensing Image Scene Classification Based on the Back Side Fusion Mode. Back side fusion combines outputs from different models, and they are fused into a single decision. This technique does not need to consider the heterogeneity of data, and each data type can utilize exclusive classifier. It is also known as decision level fusion or late fusion [39, 40, 47].

As shown in Figure 4, label_i, label_j, and label_k are classification result by CNN model_i, model_j, and model_k, respectively. They can be fused, and a final decision output label_f can be computed.

Essentially, back side fusion mode aggregates outputs from individual classifiers to constitute a final one. Many methods are proposed to manage this problem, and two commonly used back side fusion methods are listed as below.

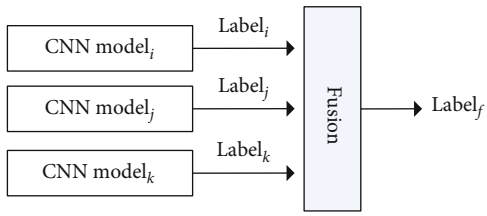

Figure 4: Demonstrations of remote sensing image scene classification based on the back side fusion mode.

3.3.1. Fusion Based on Average. Given a remote sensing image I, the classification result based on the average back side fusion method is computed as equation (4).

$$p(I) = \frac{1}{N}\sum_{i=1}^{N} p_i(I), \tag{4}$$

where $p_i()$ means ith classifier, and N is the total number. $p_i()$ gives probability value for each class label using softmax. The final result $p()$ is the value of arithmetic mean of all individual classification result [40].

3.3.2. Fusion Based on Weighted Average. Weighted average is an extension of average fusion. Weight parameters are assigned to each individual classifier [39]. It can be shown as equation (5), where w_i represents the classifier weight.

$$p(I) = \frac{1}{N}\sum_{i=1}^{N} w_i * p_i(I) \tag{5}$$

3.3.3. Using Majority Rule. In this method, the final result label_f is assigned with the one that most classifiers vote, as shown in equation (6).

$$\begin{aligned} p(e) &= \sum_{i=1}^{N} S(C_i(I) = e), \\ \text{label}_f &= \text{argmax}_j p\left(e_j\right), \end{aligned} \tag{6}$$

where C_i is ith classifier model. $S(C_i(I) = e) = 1$, if C_i classifies image I as e.

3.3.4. Using Borda Count. The Borda count for class type e is the total number of classes that ranked below e by all classifiers, as shown in equation (7).

$$\begin{aligned} p(e) &= \sum_{i=1}^{N} B_i(e), \\ \text{label}_f &= \text{argmax}_j p\left(e_j\right). \end{aligned} \tag{7}$$

$B_i(e)$ is the number of class type that ranked below e by ith classifier. The class label with maximal Borda count is regarded as the final result label_f.

4. Experiment Evaluation

To evaluate the performance and effectiveness of methods studied in this research, two widely used datasets UC Merced Land use dataset [50], WHU-RS19 dataset [51], and NWPU-RESISC45 dataset [1] are adopted in our experiments.

4.1. Dataset Description. UC Merced Land dataset is manually extracted from the USGS National Map Urban Area Imagery collection in various urban areas around the country. The pixel resolution of images in this public dataset is 1 foot. There are totally 21 classes, including agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts. 100 images are contained in each class with size of 256×256 pixels.

The images in WHU-RS19 dataset are collected from satellite images using Google Earth. The pixel resolution of images in this public dataset is about 1.6 foot. There are totally 19 classes, including airport, beach, bridge, commercial area, desert, farmland, football field, forest, industrial area, meadow, mountain, park, parking, pond, port, railway station, residential area, river, and viaduct. 50 images are contained in each class with size of 600×600 pixels.

NWPU-RESISC45 dataset is also extracted with Google Earth. It contains 31,500 images, covering 45 scene classes, including airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, intersection, island, lake, meadow, medium residential, mobile home park, mountain, overpass, palace, parking lot, railway, railway station, rectangular farmland, river, roundabout, runway, seaice, ship, snowberg, sparse residential, stadium, storage tank, tennis court, terrace, thermal power station, and wetland. 700 images are contained in each class with size of 256×256 pixels.

These three datasets are public available and designed for research purposes.

4.2. Experimental Setup. In this study, fusion-based remote sensing image scene classification methods are evaluated. There are two key testings in our experiments. First, remote sensing image scene classification based on the single side fusion mode is tested (based on front side fusion mode, middle side fusion mode, and back side fusion mode). Second, remote sensing image scene classification based on the hybrid side fusion mode is tested (combination of single side fusion mode). We randomly split the datasets into two parts. 50 % is used for model training, and the rest is used for testing. This ratio is used in all experiments. Ten experiment repetitions are conducted, each having different set division. The average performance is used as final result.

We implement all the source codes. Python is used as programming language. Tensorflow and Keras are adopted as deep learning framework and library. All experiments

are tested on Pentium i5-8 series CPU, 32G RAM, Nvidia GTX 2080Ti GPU, Ubuntu OS PC.

4.3. Performance Evaluation of the Single Side Fusion Mode. In this subsection, CNN models trained with single side fusion mode are evaluated. It is further divided into front side fusion mode, middle side fusion mode, and back side fusion mode.

For the front side fusion mode, VGG16 is used. and three methods are adopted.

(a) Front side fusion with random crop of input image. Each input image sample is first randomly cropped into 4 patches with proportion of 80 % and then resized to the original image size. The original image samples are expanded for 5 times

(b) Front side fusion with multiple scales of input image. We select 3 scales for image sample generation. There are 3 generated image samples for each raw image, with scales of 50 %, 70 %, and 90 % and then resized to the original image size. The original image samples are expanded for 4 times

(c) Front side fusion with channel combination of input image sample. Red, green, blue, and gray are used as basic channels. 3 combinations of 4 basic channels are selected, and there are 4 combined image samples generated from each training image

For middle side fusion mode, there are two methods.

(a) Single backbone. VGG16 is used as the backbone network. Feature maps of 5 maxpool layers are concatenated together. For each feature map, 1×1 conv and global average pooling(GAP) operations are used. Then, there are totally 21×5 dimensions for fused feature

(b) Multiple backbones. VGG16, Mobilenet and Resnet50 are used as backbone networks. Feature maps of the last layer of three networks are concatenated together. Softmax and MKL are then trained, respectively

For the back side fusion mode, average-based and weighted-average based methods are adopted. Multiple CNN models are trained separately (VGG16, Mobilenet and Resnet50 are used). Then, the softmax results of each model are averaged or weighted averaged, and the final result can be computed. For the weighted average-based method, the weight is trained separately.

Table 1 gives the overall accuracy and standard deviation of the single side fusion mode on UC Merced and WHU-RS19 datasets. The 1st column denotes 3 types of single side fusion mode and some baseline methods. Each mode is further divided into corresponding fusion method as described above, which is listed in the 2nd column. The accuracy results are shown in the 3rd and 4th columns.

For experiment results on the UC Merced dataset, it is shown that the back side fusion mode with average method gets the best accuracy, with value of 92.85. Meanwhile, the back side fusion mode with the weighted average method and the front side fusion mode with the multiple scale method get similar performance, with values of 92.7 and 92.68. The back side fusion mode trains multiple independent CNN models, and this way essentially incorporates multiple decisions. It has a complementary effect on classification performance. Besides, methods of the front side fusion mode get better performance than those of the middle side fusion mode. The middle side fusion mode with the multiple backbone network method gets the lowest accuracy, with value of 90.74, and the gap is about 2 % compared with the best one. For the WHU-RS19 dataset, the back side fusion mode with the weighted average method gets best accuracy, with value of 89.84. The middle side fusion mode with the multiple backbone network method gets the lowest accuracy, with value of 86.39. Front side fusion methods as a whole are better than middle side fusion methods. The gap is about 3 % between the best model and worst model. For NWPU-RESISC45 dataset, there are some differences. The back side fusion mode gets the best performance, while the middle side fusion mode gets the worst performance. The gap between different fusion modes is relatively small. Some baseline methods are also evaluated. Overall, accuracies of VGG16, Mobilenet, and Resnet50 are similar. Resnet50 gets relatively higher value, with 88.87, 84.79, and 82.78 for three datasets, while they are obviously lower than fusion based methods.

From the result, we can conclude that the back side fusion mode with the average or weighted average method adopts decisions of multiple independent models and makes these decisions as complementary and therefore, the classification performance can be improved. Front side fusion methods always get better performance than those of the middle side fusion methods. This is mainly because the front side fusion mode takes best use of the sampling technique so the size of training images can be increased to some extent. Moreover, the middle side fusion mode is not a good choice compared with others. This demonstrates that the merged heterogeneous feature with the middle side fusion mode cannot make the representation optimal. Moreover, this can also be confirmed from the results of the baseline method. Model Resnet50 has a similar structure of feature fusion, but there is no superior performance compared with the VGG16 model. Moreover, evaluations on NWPU-RESISC45, which is more large and diverse, show that the gap is smaller between different fusion modes. This demonstrates that training with preferable dataset can partly compensate model defect.

Figure 5 gives the training efficiency of single side fusion methods on UC Merced dataset. Curves of testing accuracy versus training epoch are demonstrated. The back side fusion method needs to train multiple models independently and fuse the result at decision level. The middle side fusion mode with multiple backbone (MKL) treats the feature representation and classifier construction as two separate stages. Therefore, these two methods are not end-to-end form; so, they are not included in this experiment.

As can be seen obviously in Figure 5, training with the front side fusion mode is more stable while training with

Table 1: Overall accuracies (%) and standard deviations of the single side fusion mode on UC Merced, WHU-RS19, and NWPU-RESISC45 datasets.

Mode	Method	UC Merced	WHU-RS19	NWPU-RESISC45
Front side	Random crop	91.54 ± 0.72	88.24 ± 0.43	87.82 ± 0.37
	Multiscale	92.68 ± 0.65	88.78 ± 0.55	87.6 ± 0.29
	Multichannel	91.43 ± 0.68	87.71 ± 0.58	86.19 ± 0.84
Mid side	Single backbone	90.95 ± 0.73	87.52±0.29	86.76 ± 0.19
	Multibackbone	90.74 ± 0.45	86.39 ± 0.65	86.84 ± 0.32
	Multibackbone(MKL)	91.38 ± 0.37	87.47 ± 0.67	86.67 ± 0.53
Back side	Average	92.85 ± 0.76	89.67 ± 0.36	88.23 ± 0.29
	Weighted average	92.7 ± 0.56	89.84 ± 0.49	88.18 ± 0.84
Baseline method	VGG16	88.76 ± 0.63	84.76 ± 0.39	82.3 ± 0.31
	Mobilenet	87.12 ± 0.74	83.35 ± 0.27	81.21 ± 0.23
	Resnet50	88.87 ± 0.57	84.79 ± 0.51	82.78 ± 0.59

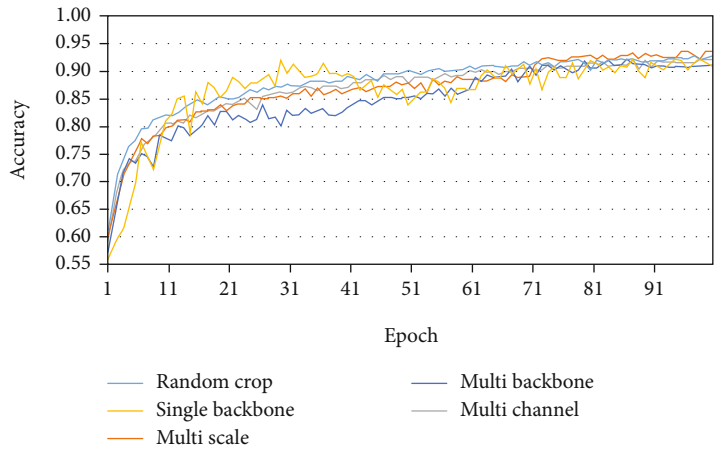

Figure 5: Training efficiency of single side fusion mode.

the middle side fusion mode is more fluctuate. The middle side fusion mode with the single backbone method combines feature from different levels of the network. The middle side fusion mode with the multiple backbone method combines feature from different CNN networks. So, the parameter tuning in these two methods is more difficult, and the models are relatively hard to train. Comparatively, the front side fusion mode adopts the same CNN network structure so the model training is relatively easy and is more stable. Similar results can be gained on the other two datasets, which are omitted here.

4.4. Performance Evaluation of the Hybrid Side Fusion Mode. In this subsection, CNN models trained with the hybrid side fusion mode are evaluated, which is designed with more than one side fusion mode. It is further divided into front+back side, front+mid side, mid + back side, and front+mid + back side fusion modes.

Table 2 gives the overall accuracy and standard deviation of hybrid side fusion methods on UC Merced, WHU-RS19, and NWPU-RESISC45 datasets. The 1st column denotes four hybrid side fusion modes. The combinations of corresponding various single side fusion methods are listed in the 2nd column. There are totally 24 models to be trained. The accuracy results are shown in the 3rd, 4th, and 5th columns. For convenience, average (w) is used to represent weighted average.

Take the result on UC Merced dataset as example. For the front+back side fusion mode, the multiscale+average method gets the best performance, 96.94. There is a slight gap compared with other methods (random crop+average or (w) average, multichannel+average or average (w)). This demonstrates the effectiveness of the back side fusion mode. For the front +middle side fusion mode, the multiscale+single backbone method gets the best performance, 96.19. The multichannel +multibackbone (MKL) method gets the worst performance, 94.49. It can also be seen that the single backbone method can better promotes performance of the front side fusion mode. For the mid + back side fusion mode, all methods get similar performance, with values range from 96.4 to 96.7. This

Table 2: Overall accuracies (%) and standard deviations of hybrid side fusion on UC Merced, WHU-RS19, and NWPU-RESISC45 datasets.

Mode	Method	UC Merced	WHU-RS19	NWPU-RESISC45
Front + back	Random crop+average	96.85 ± 0.36	92.75 ± 0.51	94.93 ± 0.87
	Multiscale+average	96.94 ± 0.41	93.14 ± 0.39	94.85 ± 0.42
	Multichannel+average	96.51 ± 0.52	91.91 ± 0.47	94.07 ± 0.63
	Random crop+average (w)	96.75 ± 0.31	92.71 ± 0.46	94.9 ± 0.56
	Multiscale+average(w)	96.84 ± 0.48	93.19 ± 0.23	94.87 ± 0.66
	Multichannel+average (w)	96.65 ± 0.45	92.03 ± 0.52	94.19 ± 0.24
Front + mid	Random crop+single backbone	96.15 ± 0.29	91.35 ± 0.51	93.46 ± 0.57
	Random crop+multibackbone	95.65 ± 0.62	90.59 ± 0.42	93.4 ± 0.32
	Random crop+multibackbone (MKL)	95.78 ± 0.75	91.02 ± 0.58	93.14 ± 0.27
	Multiscale+single backbone	96.19 ± 0.32	91.59 ± 0.62	93.56 ± 0.52
	Multiscale+multibackbone	95.34 ± 0.47	91.04 ± 0.29	93.25 ± 0.61
	Multiscale+multibackbone (MKL)	95.41 ± 0.53	90.8 ± 0.43	92.89 ± 0.62
	Multichannel+single backbone	95.51 ± 0.48	91.11 ± 0.58	93.01 ± 0.27
	Multichannel+multibackbone	94.57 ± 0.51	90.48 ± 0.45	92.96 ± 0.74
	Multichannel+multibackbone (MKL)	94.49 ± 0.45	90.54 ± 0.65	92.66 ± 0.69
Mid + back	Single backbone+average	96.55 ± 0.29	92.75 ± 0.59	94.15 ± 0.36
	Multibackbone+average	96.64 ± 0.47	92.68 ± 0.14	93.95 ± 0.52
	Multibackbone (MKL) + average	96.58 ± 0.38	92.88 ± 0.57	94.07 ± 0.32
	Single backbone+average (w)	96.45 ± 0.37	92.53 ± 0.32	94.18 ± 0.42
	Multibackbone+average (w)	96.73 ± 0.56	92.54 ± 0.29	93.87 ± 0.75
	Multibackbone (MKL) + average (w)	96.41 ± 0.51	92.65 ± 0.69	93.95 ± 0.39
Front+mid + back	Random crop+single backbone+average	97.05 ± 0.45	94.85 ± 0.55	95.45 ± 0.27
	Random crop+multibackbone+average	97.14 ± 0.32	94.74 ± 0.72	95.24 ± 0.63
	Random crop+multibackbone (MKL) + average	96.86 ± 0.37	93.76 ± 0.57	95.04 ± 0.43
	Multiscale+single backbone+average	96.87 ± 0.29	94.17 ± 0.71	95.36 ± 0.54
	Multiscale+multibackbone+average	96.74 ± 0.35	94.67 ± 0.58	95.52 ± 0.64
	Multiscale+multibackbone (MKL) + average	96.16 ± 0.31	93.42 ± 0.81	94.31 ± 0.74
	Multichannel+single backbone+average	96.79 ± 0.38	94.19 ± 0.34	94.79 ± 0.56
	Multichannel+multibackbone+average	96.54 ± 0.47	94.04 ± 0.56	94.64 ± 0.51
	Multichannel+multibackbone (MKL) + average	96.63 ± 0.42	93.83 ± 0.47	94.14 ± 0.7
	Random crop+single backbone+average (w)	97.12 ± 0.41	94.47 ± 0.75	95.57 ± 0.32
	Random crop+multibackbone+average (w)	97.04 ± 0.39	94.54 ± 0.51	95.61 ± 0.47
	Random crop+multibackbone(MKL) + average (w)	96.53 ± 0.34	93.85 ± 0.46	94.63 ± 0.27
	Multiscale+single backbone+average (w)	96.89 ± 0.43	94.25 ± 0.42	95.5 ± 0.27
	Multiscale+multibackbone+average (w)	96.77 ± 0.42	94.61 ± 0.34	95.39 ± 0.68
	Multiscale+multibackbone(MKL) + average (w)	96.09 ± 0.58	93.47 ± 0.65	94.32 ± 0.74
	Multichannel+single backbone+average (w)	96.61 ± 0.19	94.15 ± 0.39	94.52 ± 0.7
	Multichannel+multibackbone+average (w)	96.13 ± 0.68	93.75 ± 0.66	94.43 ± 0.45
	Multichannel+multibackbone(MKL) + average (w)	96.24 ± 0.47	93.61 ± 0.29	94.22 ± 0.58

demonstrates that the middle side fusion mode plays less effectiveness with regard to the back side fusion mode. For the front+mid + back fusion mode, the random crop+multiple backbone+average method gets best performance of 97.14, and multiscale+multibackbone (MKL) + average gets the worst performance of 96.16. Moreover, random crop+X + average ("X" denotes middle side fusion method) methods are generally better than other methods.

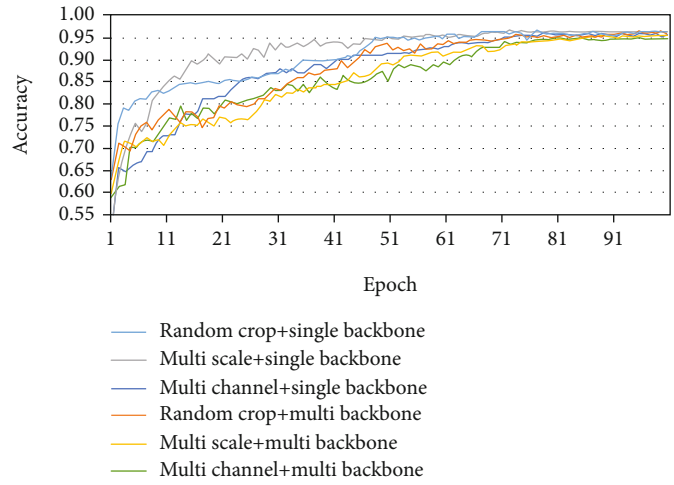

Figure 6: Training efficiency of single side feature fusion.

For comparison of the front+back side fusion mode and front+mid side fusion mode, the former is better. This indicates that the back side fusion mode is more useful than the middle side fusion mode for model performance improvement. The difference of performance between the front+back side fusion mode and middle+back side fusion mode is very small, with about 0.2 %. This indicates that the back side fusion mode is dominant for the front side fusion mode and middle side fusion mode. Front side fusion mode and middle side fusion mode are relatively unimportant for model performance boosting compared with the back side fusion mode. This may also stem from that front and middle modes play the benefit for image representation. For all fusion modes, the front+mid + back side fusion mode gets the best performance. This verifies that using all fusion techniques, including front, middle, and back side fusion, model performance is indeed increased for remote sensing image scene classification. The hybrid side fusion mode can make the best use of powerful ability for feature extraction and representation. The front+mid side fusion mode is the least efficient way compared with other modes. The essential and obvious differences stems from that it lacks the back side fusion mode, which illustrates the merit of back side fusion mode as well.

For WHU-RS19 and NWPU-RESISC45 dataset, the performance evaluation shows similar results. The front+mid + back side fusion mode works better than other fusion modes, and the front+mid side fusion mode gets the worst performance.

Figure 6 gives the training efficiency of the hybrid side fusion method. Curves of testing accuracy versus training epoch are demonstrated. Like Figure 5, fusion methods with the back side fusion mode or with multiple backbone (MKL) are not considered in this experiment.

As can be seen from Figure 6, the multiscale+single backbone method works best, and it uses minimal cost to converge. Multiscale+multibackbone and multichannel +multibackbone methods are more fluctuate and are hard to train. Single backbone is more advantageous to multiscale than the multibackbone method. This illustrates the complexity of the multibackbone method, which needs to train various inhomogeneous CNN networks simultaneously. Similar cases are demonstrated on the other two datasets, which are omitted here.

4.5. Compared with Others Methods. In this subsection, performance comparisons of other works that using fusion based method are given.

Five related fusion based methods, which are proposed in refs. [36, 37, 40, 42, 44], are used for comparison. These methods have been described in previous section. The results are given in Table 3. It can be seen that the random crop +multibackbone+average-based fusion method outperforms other methods with about 1-4 %, which demonstrates the effectiveness of our fine-tuned fusion-based method.

4.6. Performance Evaluation of Other Parameters. In this subsection, some important parameters in model training are explained, including the size of input image, optimization method, batch size, model size, and model efficiency.

4.6.1. Size of Input Image. In this testing, the random crop +multibackbone+average method is used to train the model with various sizes of input image. The results are given in Table 4.

For UC Merced dataset, training image samples are set with 128×128 pixels, 200×200 pixels, 256×256 pixels, and 300×300 pixels. Experiment evaluation shows that image samples with size of 128×128 pixels gets the best performance. It outperforms others with about 1.2 %, 0.8 %, and 1.8 %, respectively. For WHU-RS19 dataset, training image samples are set with 200×200 pixels, 300×300 pixels, 400×400 pixels, and 600×600 pixels. Experiment evaluation shows that image samples with size of 400×400 pixels get the best performance. It outperforms others with about 1.5 %, 1.1 %, and 0.8 %, respectively. For NWPU-RESISC45 dataset, training image samples are set with 128×128 pixels, 200×200 pixels, 256×256 pixels, and 300×300 pixels. Experiment evaluation shows that image samples with size

Table 3: Performance comparison with other methods.

Mode	UC Merced	WHU-RS19	NWPU-RESISC45
Method in [36]	96.18 ± 0.75	93.07 ± 0.79	91.52 ± 0.4
Method in [37]	95.76 ± 0.75	92.78 ± 0.35	91.18 ± 0.85
Method in [40]	95.59 ± 0.48	92.93 ± 0.56	91.34 ± 0.47
Method in [42]	94.50 ± 0.18	92.21 ± 0.55	90.84 ± 0.54
Method in [44]	96.37 ± 0.43	93.52 ± 0.45	92.95 ± 0.67
Random crop+multibackbone+average	97.14 ± 0.32	94.74 ± 0.72	94.14 ± 0.7

Table 4: Overall accuracies (%) and standard deviations of various input image sizes on UC Merced, WHU-RS19, and NWPU-RESISC45 datasets.

Dataset	Training image size	OAs
UC Merced	128 × 128 pixels	97.14 ± 0.32
	200 × 200 pixels	95.33 ± 0.56
	256 × 256 pixels	96.12 ± 0.45
	300 × 300 pixels	96.34 ± 0.61
WHU-RS19	200 × 200 pixels	93.23 ± 0.52
	300 × 300 pixels	93.63 ± 0.39
	400 × 400 pixels	94.74 ± 0.72
	600 × 600 pixels	93.94 ± 0.63
NWPU-RESISC45	128 × 128 pixels	94.68 ± 0.52
	200 × 200 pixels	95.14 ± 0.56
	256 × 256 pixels	95.24 ± 0.67
	300 × 300 pixels	95.02 ± 0.71

of 256 × 256 pixels get the best performance. It outperforms others with about 0.5 %, 0.1 %, and 0.2 %, respectively.

4.6.2. Optimization Method. For the optimization method, SGD, Adam, and Adagrad are evaluated. For comparison, random crop+multibackbone is used as the basic fusion method, and other parameters are set the same as Section 3.4. Among the above optimization methods, SGD is stable for training models. While for Adam and Adagrad, it illustrates that they cannot always keep smooth and steady. They are relatively hard to make the model convergence, which makes the training procedure more costly.

4.6.3. Batch Size for Model Training. Batch size is another important parameter for model training. In our research, batch size with 16, 32, 48, and 64 are evaluated. For comparison, random crop+multibackbone is used as the basic fusion method, and other parameters are set the same as Section 3.4. The experiment result shows that various batch sizes have less impact on classification accuracy of models, and the gap lies in 0.3 % It makes some influence on model training efficiency, and models with batch size 16 are easy and fast to train.

4.6.4. Model Size and Efficiency. In this testing, times for training and testing of each method are evaluated. Meanwhile, the numbers of weight parameters of each method are also reported. UC Merced dataset is used for demonstration.

Table 5 gives the evaluations of model size and efficiency of each method. The 1st and 2nd columns denote the fusion mode and method. The 3rd column gives the model size (number of weight parameters), which is measured with million. The 4th and 5th column are times for model training and testing. Training time represents the cost for one training step with batchsize 32.

For middle side fusion with multiple backbone and back side fusion with average, three CNN networks, VGG16, Resnet50, and Mobilenet, are used. So, the corresponding model sizes are all approximately 168 M. For the front side +single backbone method, the model size is approximately 138 M (We use VGG16 as single backbone network).

For front side fusion+single backbone methods, training and testing times are about 5.16 s and 0.17 s, respectively. For front side fusion+multibackbone methods, training and testing times are about 8.73 s and 0.26 s, respectively. It is more time consuming because three CNN models are combined, and more computations are needed. For front side fusion+middle side fusion+average methods, training and testing times are about 12.6 s and 0.35 s. They take the most time for three CNN models that need to be trained separately (it takes the same time for all methods combined with the back fusion mode.).

Back side fusion with average and weighted average costs the same computation and model size, and so the weighted average method is omitted. Fusion with MKL-based methods is not considered in this experiment, for it needs special libraries and cannot be evaluated on the same conditions (actually, training MKL costs quite long time).

5. Discussion

Remote sensing image scene classification is an important task for geographical analysis and region monitor. Currently, fusion-based methods are most commonly used for their excellent performance.

Among all fusion modes, back side with average is the most effective and convenient way, while the shortcoming is that it needs to training multiple independent CNN models, which may cost more resource for model training

Table 5: Model size and efficiency.

Mode	Method	Size (M)	Training (s)	Testing (s)
Front + back	Random crop+average	168	12.6	0.35
	Multiscale+average	168	12.6	0.35
	Multichannel+average	168	12.6	0.35
Front+mid	Random crop+single backbone	138	5.16	0.17
	Random crop+multibackbone	168	8.73	0.26
	Multiscale+single backbone	138	5.16	0.17
	Multiscale+multibackbone	168	8.73	0.26
	Multichannel+single backbone	138	5.16	0.17
	Multichannel+multibackbone	168	8.73	0.26
Mid + back	Single backbone+average	168	12.6	0.35
	Multibackbone+average	168	12.6	0.35
Front+mid + back	Random crop+single backbone+average	168	12.6	0.35
	Random crop+multibackbone+average	168	12.6	0.35
	Multiscale+single backbone+average	168	12.6	0.35
	Multiscale+multibackbone+average	168	12.6	0.35
	Multichannel+single backbone+average	168	12.6	0.35
	Multichannel+multibackbone+average	168	12.6	0.35

and more expensive on computation and storage. Therefore, many researches do not pay much attentions on back fusion mode in practice and real application [36, 37, 42, 44].

For the front side fusion mode, random crop and multiscale are more effective methods [36, 37, 40]. Essentially, the front side fusion mode can be seen as training data augmentation, which is a universal technique and can be used in all kinds of model training.

Middle side fusion is currently the research spot. Special architectures and complex operations are designed [42, 44]. However, in our evaluation, front side fusion mode and middle side fusion mode play similar purpose, and some methods may be offset to each other. In some cases, small cumulative effect is gained, and even duplicated objectives may take place. Moreover, single backbone and multibackbone methods are preferred to the MKL method. MKL method does not show superiority, and its training is inefficient for there is not enough libraries supporting fast parallel computing.

The aim of fusion is to learn image representation on multiple levels and degrees for key factors in scene image may appear with various scales. Fusion operations are widely adopted in state of art research. Various fusion modes (front side, middle side, and back side) are often combined to boost the model performance. In this paper, we only study common fusion methods for remote sensing image scene image classification. General trends are demonstrated by experimental evaluation. Other complex fusion methods are difficult to categorize and evaluated, and so they are not our focus and not included in this work.

6. Conclusions

This paper makes a research on the fusion-based method for remote sensing image scene classification. Three types of fusion modes, front side fusion, middle side fusion, and back side fusion, are defined. Typical methods for different fusion modes are given. Comprehensive experiments are carried out. Combinations of various fusion modes are evaluated. Results of model accuracy and training efficiency on commonly used datasets are given. It demonstrates that the random crop+multiple backbone+average method has the best performance. Characteristics of different fusion modes and their interaction are analyzed.

Our future works will focus on two aspects: (1) we will make more indepth research on the fusion-based method with specific structure. (2) External dataset should be used to improve model performance, such as webly grabbed form image engine or Google maps, for they can provide dominant training image samples.

Data Availability

The data supporting this research are from previously reported studies and datasets, which have been cited. The processed data are available at http://weegee.vision.ucmerced.edu/datasets/landuse.html, http://www.escience.cn/people/yangwen/whu-rs19.html, and https://1drv.ms/u/s!AmgKYzARBl5ca3HNaHIlzp_IXjs.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was supported by the Fundamental Research Funds for the Central Universities of China (Northeastern University with No. N2017007).

References

[1] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[2] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[3] H. Li, H. Gu, Y. Han, and J. Yang, "Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine," *International Journal of Remote Sensing*, vol. 31, no. 6, pp. 1453–1470, 2010.

[4] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3706–3715, 2006.

[5] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 3023–3034, 2014.

[6] B. Cui, X. Xie, X. Ma, G. Ren, and Y. Ma, "Superpixel-based extended random walker for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3233–3243, 2018.

[7] L. Huang, C. Chen, W. Li, and Q. du, "Remote sensing image scene classification using multi-scale completed local bBinary patterns and fisher vectors," *Remote Sensing*, vol. 8, no. 6, p. 483, 2016.

[8] S. Banerji, A. Sinha, and C. Liu, "A new bag of words LBP (BoWL) descriptor for scene image classification," *Computer Analysis of Images and Patterns*, pp. 490–497, 2013.

[9] Y. Yang and S. D. Newsam, "Comparing SIFT descriptors and gabor texture features for classification of remote sensed imagery," in *2008 15th IEEE International Conference on Image Processing*, pp. 1852–1855, San Diego, CA, USA, 2008.

[10] J. Lei, K. Xie, H. Zheng, B. Zhang, and W. Yang, "GPCA-SIFT: A New Local Feature Descriptor for Scene Image Classification," in *Pattern Recognition. CCPR 2016. Communications in Computer and Information Science*, T. Tan, X. Li, X. Chen, J. Zhou, J. Yang, and H. Cheng, Eds., vol. 2, pp. 286–295, Springer, Singapore, 2016.

[11] S. Qi, J. Ma, J. Lin, Y. Li, and J. Tian, "Unsupervised ship detection based on saliency and S-HOG descriptor from optical satellite images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 1451–1455, 2015.

[12] G. Cheng, P. Zhou, X. Yao, C. Yao, Y. Zhang, and J. Han, "Object detection in VHR optical remote sensing images via learning rotation-invariant HOG feature," in *2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*, pp. 433–436, Guangzhou, China, 2016.

[13] J. Yin, H. Li, and X. Jia, "Crater detection based on gist features," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 1, pp. 23–29, 2015.

[14] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[16] T.-Y. Lin, M. Maire, S. J. Belongie et al., "Microsoft COCO: common objects in context," in *Computer Vision – ECCV 2014. ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693 of Lecture Notes in Computer Science, pp. 740–755, Springer, Cham, 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS*, pp. 1106–1114, 2012.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014. ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8689 of Lecture Notes in Computer Science, pp. 818–833, Springer, Cham, 2014.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, pp. 1–14, 2015.

[20] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[22] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, "Using convolutional features and a sparse autoencoder for land-use scene classification," *International Journal of Remote Sensing*, vol. 37, no. 10, pp. 2149–2167, 2016.

[23] X. Han, Y. Zhong, B. Zhao, and L. Zhang, "Scene classification based on a hierarchical convolutional sparse auto-encoder for high spatial resolution imagery," *International Journal of Remote Sensing*, vol. 38, no. 2, pp. 514–536, 2017.

[24] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Transactions on Geoscience Remote Sensing*, vol. 54, no. 6, pp. 3360–3671, 2016.

[25] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, "Remote sensing image scene classification using bag of convolutional features," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1735–1739, 2017.

[26] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.

[27] J. Chen, C. Wang, Z. Ma, J. Chen, D.-x. He, and S. Ackland, "Remote sensing scene classification based on convolutional neural networks pre-trained using attention-guided sparse filters," *Remote Sensing*, vol. 10, p. 290, 2018.

[28] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 1155–1167, 2019.

[29] J. Fang, Y. Yuan, X. Lu, and Y. Feng, "Robust space-frequency joint representation for remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7492–7502, 2019.

[30] G. Chen, X. Zhang, X. Tan et al., "Training small networks for scene classification of remote sensing images via knowledge distillation," *Remote Sensing*, vol. 10, no. 5, p. 719, 2018.

[31] L. Bashmal, Y. Bazi, H. AlHichri, M. M. AlRahhal, N. Ammour, and N. Alajlan, "Siamese-GAN: learning invariant representations for aerial vehicle image categorization," *Remote Sensing*, vol. 10, no. 3, p. 351, 2018.

[32] S. Xu, X. Mu, D. Chai, and X. Zhang, "Remote sensing image scene classification based on generative adversarial networks," *Remote Sensing Letters*, vol. 9, pp. 617–626, 2018.

[33] D. Ma, P. Tang, and L. Zhao, "SiftingGAN: generating and sifting labeled samples to improve the remote sensing image scene classification BaselineIn vitro," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 7, pp. 1046–1050, 2019.

[34] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020.

[35] 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 Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019.

[36] Q. Liu, R. Hang, H. Song, and Z. Li, "Learning multiscale deep features for high-resolution satellite image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 117–126, 2018.

[37] Y. Liu, Y. Zhong, and Q. Qin, "Scene classification based on multiscale convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7109–7121, 2018.

[38] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 183–186, 2018.

[39] Y. Yu and F. Liu, "Aerial scene classification via multilevel fusion based on deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 287–291, 2018.

[40] N. Yang, H. Tang, H. Sun, and X. Yang, "DropBand: a simple and effective method for promoting the scene classification accuracy of convolutional neural networks for VHR remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 257–261, 2018.

[41] U. Muhammad, W. Wang, S. P. Chattha, and S. Ali, "Pre-trained VGGNet architecture for remote-sensing image scene classification," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1622–1627, Beijing, China, 2018.

[42] G. Wang, B. Fan, S. Xiang, and C. Pan, "Aggregating rich hierarchical features for scene classification in remote sensing imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 9, pp. 4104–4115, 2017.

[43] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775–4784, 2017.

[44] X. Lu, H. Sun, and X. Zheng, "A feature aggregation convolutional neural network for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7894–7906, 2019.

[45] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5148–5157, 2017.

[46] X. Lu, W. Ji, X. Li, and X. Zheng, "Bidirectional adaptive feature fusion for remote sensing scene classification," *Neurocomputing*, vol. 328, pp. 135–146, 2019.

[47] F. Zhang, B. du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1793–1802, 2016.

[48] R. Minetto, M. Pamplona Segundo, and S. Sarkar, "Hydra: an ensemble of convolutional neural networks for geospatial land classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6530–6541, 2019.

[49] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, no. 3, pp. 2491–2521, 2008.

[50] Y. Yang and S. D. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, pp. 270–279, 2010.

[51] G. S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Maitre, and H. Sun, "Structural high-resolution satellite image indexing," in *Symposium: 100 Years ISPRS - Advancing Remote Sensing Science*, Vienna, Austria, 2010.