

Received March 1, 2022, accepted March 24, 2022, date of publication March 30, 2022, date of current version April 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163256

Improving Bag-of-Deep-Visual-Words Model via Combining Deep Features With Feature Difference Vectors

XIANGSHI WANG 

School of IoT Technology, Wuxi Institute of Technology, Wuxi 214121, China


e-mail: xswang_wxit@163.com

ABSTRACT Bag-of-Deep-Visual-Words (BoDVW) model has shown its advantage over Convolutional Neural Network (CNN) model in image classification tasks with a small number of training samples. An essential step in BoDVW model is to extract deep features by using an off-the-shelf CNN model as a feature extractor. Two deep feature extraction methods have been raised in recent years. The first method densely samples multi-scale image patches and then converts them into deep features via a deep-level fully-connected layer. The second method uses the output of a deep-level convolutional layer or pooling layer as the source of deep features. By contrast, the second method is much more efficient. However, it performs worse than the first method in classification accuracy. The reason is that deep features extracted by the second method are yielded in receptive fields of a single size. To make BoDVW model have high feature extraction efficiency and high classification accuracy, we propose enhancing deep features extracted by the second method at low added computation costs by supplementing the information obtained from receptive fields of different sizes. Concretely, we raise a novel feature named “feature difference (FD) vector” in this article. It can roughly preserve the information of multiple deep features extracted by the convolutional layers of different receptive field sizes. Each deep feature is enhanced by combining an FD vector to form a combined feature. The image representation vector of an image is generated using the combined features extracted from it. Our experimental results on three public datasets (15-Scenes, TF-Flowers, and NWPU-RESISC45) show that our method can avoid the high computation costs of the first method and achieve comparable results to the first method, which exhibits the effectiveness of our method.

INDEX TERMS Image classification, bag-of-deep-visual-words, feature extraction, deep feature, feature difference.

I. INTRODUCTION

Image classification, as a key problem in computer vision, has attracted much attention for a long time. In a decade, with the necessary support of a large number of training samples and rich computing resources, Convolutional Neural Network (CNN) model has exhibited its significant performance in many challenging classification tasks, such as the large-scale competition of ImageNet classification in 2012 (ILSVRC-12) [1]. Besides CNN model, Bag-of-Visual-Words (BoVW) model [2], and Bag-of-Deep-Visual-Words (BoDVW) model [3], [4] also play their roles in some tasks with a small number of training samples.

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang .

BoDVW model is a combination product of BoVW model and CNN model. It has almost the same workflow as BoVW model, including feature extraction, dictionary learning, feature coding, feature pooling, and classifying [5]. The only difference is the way of feature extraction. BoVW model calculates handcrafted features (e.g., Scale-Invariant Feature Transform (SIFT) and Histogram of Gradient (HoG)) over small image patches (e.g., 16×16 pixels) [5]. In contrast, BoDVW model extracts features via an off-the-shelf CNN model pre-trained on a large-scale dataset [3], [4]. Since features extracted by BoDVW model possess richer semantic information and larger receptive fields than that of BoVW model, they are named “deep features” in the literature. Owing to deep features, BoDVW model has shown its advantage over CNN model on some small datasets such as NWPU-RESISC45 [6], TF-Flowers [7], COVID-19 [8] and so on.

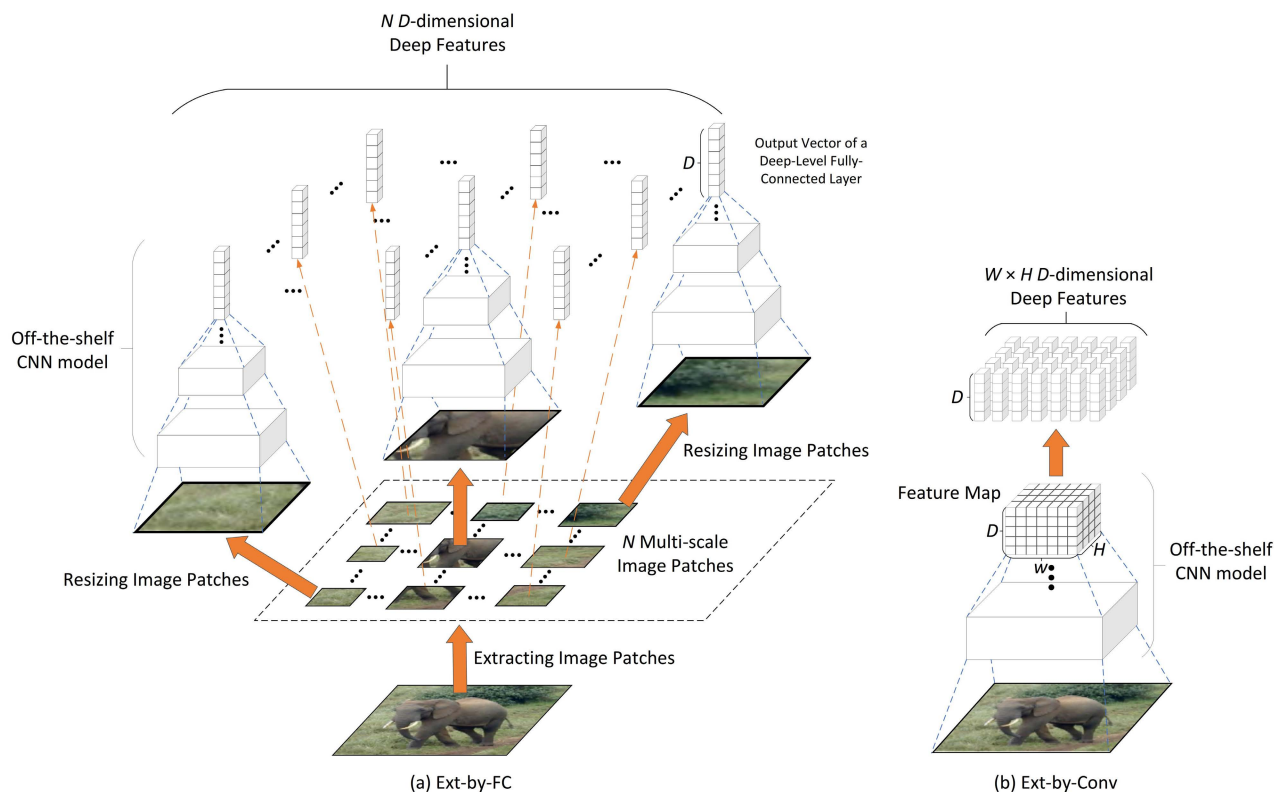


FIGURE 1. Two deep feature extraction methods. (a) Ext-by-FC: extracting deep features via a deep-level fully-connection layer; (b) Ext-by-Conv: extracting deep features via a deep-level convolutional layer or pooling layer.

There are two methods for extracting deep features in the literature, as shown in Figure 1. The first one (denoted as **Ext-by-FC**) densely samples multi-scale image patches (e.g., 128×128 pixels, 160×160 pixels, 192×192 pixels) from an image [3], [4], [9], [10]. Each patch is resized to fit the input size (e.g., 224×224 pixels for ResNet-50) of an off-the-shelf CNN model, then the output of a deep-level fully-connected layer for each resized patch is regarded as a deep feature. The second one (denoted as **Ext-by-Conv**) directly takes the output of a deep-level convolutional layer or pooling layer as the source of deep features [11]–[13]. Providing that the output size of a convolutional layer is $W \times H \times D$ where D is the number of feature maps, then $W \times H$ D -dimensional deep features will be generated for each image.

By contrast, as shown in Section V(D), Ext-by-Conv performs worse than Ext-by-FC in classification accuracy. The reason is that deep features extracted by a convolutional layer or pooling layer are generated in receptive fields of a single size. In contrast, deep features converted from multi-scale image patches can be viewed as being generated in receptive fields of different sizes. However, Ext-by-FC is a time-consuming extraction method since the inference of the used CNN model is performed one time for each image patch. In comparison, Ext-by-Conv is much more efficient because it only needs to perform the inference one time to obtain all deep features.

To make BoDWW model have high feature extraction efficiency and high classification accuracy, we propose enhancing deep features extracted by Ext-by-Conv at a low added computation costs by supplementing the information obtained from receptive fields of different sizes. Concretely, considering that different convolutional layers have different receptive field sizes, a novel feature named “feature difference (FD) vector” is proposed to preserve the information of multiple deep features extracted by different convolutional layers. An FD vector records the differences among multiple deep features at a location. Each of the deep features extracted at the deepest-level convolution layer is enhanced by combining an FD vector to form a combined feature. The image representation vector of an image is generated using the combined features extracted from it.

In our experiments, two popular off-the-shelf CNN models, VGGNet-16 [1] and ResNet-50 [14] (implemented by PyTorch), are used as feature extractors. A representative coding method is chosen for encoding combined features, i.e., locality-constrained linear coding (LLC) [16]. Our method is evaluated on three public image datasets, including 15-Scenes [15], TF-Flowers [16], and NWPU-RESISC45 [9]. Experimental results show that our method can achieve comparable results to Ext-by-FC, and the computation costs spent on feature extraction are much lower than that of Ext-by-FC. These results exhibit the effectiveness of our method.

The contribution of our work is that we propose a feature enhancement method to make BoDVW model have both high feature extraction efficiency and high classification accuracy. Our method can avoid the high computation costs of Ext-by-FC and achieve comparable results to Ext-by-FC. In addition, a novel feature named “FD vector” is raised in this article. It can be considered to combine with other features such as handcrafted features to solve other computer vision problems.

The remainder of this article is organized as follows: the proceeding section is about the related works. Section III illustrates the workflow of our method. Section IV explains our method in detail. Experimental evaluation and analysis are reported in Section V. The discussion is presented in Section VI, and the conclusion is drawn in Section VII.

II. RELATED WORKS

The related works to our work can be summarized from three aspects, including workflow, feature extraction, and feature fusion.

From the aspect of workflow, our method shares almost the same workflow as BoVW model. Huang *et al.* [5] had concluded the general workflow of BoVW model in 2014. It can be divided into five stages, i.e., feature extraction, feature coding, dictionary learning, feature pooling, and classifying. A large number of works have been devoted to improving the classification performance of BoVW model from one or two of the five aspects before 2015 [17]–[20]. Especially, there are many coding methods proposed in the era of BoVW model [5], e.g., hard voting, soft voting, sparse coding, LLC, local coordinate coding, super vector coding, fisher coding, grouping saliency coding, etc. Given the high similarity between BoVW model and BoDVW model in terms of workflow, some of them have been applied in the existing BoDVW methods such as hard voting [12], [13], sparse coding [4], and LLC [10].

From the aspect of feature extraction, FD vectors are generated based on deep features. Two deep feature extraction methods have been proposed in the literature. The first one (Ext-by-FC) takes the outputs of a deep-level fully-connected layer for multi-scale image patches as deep features. In [10], image patches of 128×128 pixels, 92×92 pixels, and 64×64 pixels are densely sampled from each image and then transformed into deep features via the last fully-connected layer of AlexNet. In [3], image patches of 256×256 pixels, 224×224 pixels, 192×192 pixels, 160×160 pixels, and 128×128 pixels are converted into deep features via the last fully-connected layer of DeCAF₆. Other similar works include [4] and [9]. The second one (Ext-by-Conv) takes the output of a deep-level convolutional layer or pooling layer as the source of deep features. In [11], the author adopted the layer “conv5” of AlexNet, the layer “inception 4(e)” of GoogleNet, and the layer “conv5-3” of VGGNet-16 to extract features, leading to 169 256-dimensional features, 196 832-dimensional features, and 196 512-dimensional features for each image, respectively. In [12], the last convolutional

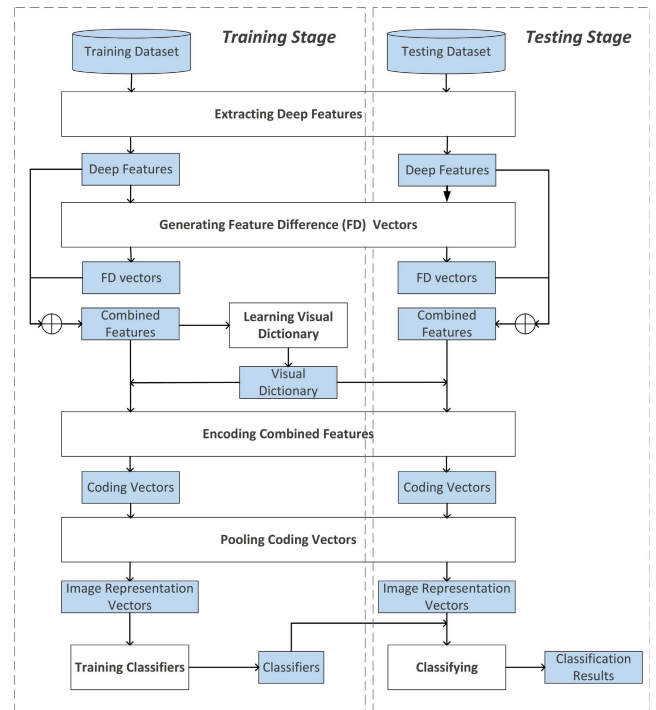


FIGURE 2. Workflow of the BoDVW model improved by our method.

layer of ResNet-50 is used to generate 49 2048-dimensional features for each image.

From the aspect of feature fusion, each deep feature is combined with an FD vector to form a combined feature. This practice is a typical way of feature fusion. Deep feature fusion is also a flourishing research area. The recent works in this area can be grouped into three groups. The first group combines deep features with handcrafted features to solve task-specific classification problems as done in [21], [22]. The second group [23]–[25] adds new branches to multiple intermediate layers of an off-the-shelf model to aggregate the outputs of these layers by fine-tuning the modified model. The third group incorporates the outputs of different CNN models or the outputs of different layers of a CNN model by a certain algorithm, such as metric learning [26], sparse representation learning [27], discriminant correlation analysis [28], etc.

III. WORKFLOW

The workflow of the BoDVW model improved by our method is shown in Figure 2. It includes the training stage and the testing stage.

The first step at the training stage utilizes an off-the-shelf CNN model as a feature extractor to extract deep features. The outputs of multiple convolutional layers are used as the source of deep features. Then, FD vectors are yielded based on extracted deep features. Next, each of the deep features obtained at the deepest-level convolutional layer is combined with an FD vector to generate a combined feature. Afterward, combined features obtained from training samples are used

TABLE 1. Abbreviations used in this article and their meanings.

Abbreviation	Meaning
BoVW	bag-of-visual-words
BoDVW	bag-of-deep-visual-words
CNN	convolutional neural network
FD	feature difference
Ext-by-FC	feature extraction method that extracts deep feature via a fully-connected layer
Ext-by-Conv	feature extraction method that extracts deep features via a convolutional layer or pooling layer
SPP	spatial pyramid pooling
LLC	locality-constrained linear coding
ResNet-50(no-refined)	fine-tuned ResNet-50 of which only the fully-connected part is re-trained
ResNet(refined)	fine-tuned ResNet-50 of which all layers are fine-tuned
VGGNet-16(no-refined)	fine-tuned VGGNet-16 of which only the fully-connected part is re-trained
VGGNet-16(refined)	fine-tuned VGGNet-16 of which all layers are fine-tuned
D	dimensionality of deep features
L	number of the convolutional layers used to generate FD vectors
G	number of the sub-vectors into which a deep feature is divided

to learn a visual dictionary. In the next step, each combined feature is encoded as a coding vector with the learned dictionary. After attaining the coding vectors of an image, the image representation vector is yielded by pooling together all the coding vectors. At last, the image representation vectors of all training samples are used to train classifiers.

The workflow of the testing stage is similar to that of the training stage. The differences are that for each testing image, the combined features extracted from it are encoded with the dictionary learned at the training stage, and its representation vector is fed into the classifiers trained at the training stage to obtain its predicted category.

IV. OUR METHOD

An essential step in BoDVW model is to extract deep features via an off-the-shelf CNN model as a feature extractor. There have been two deep feature extraction methods Ext-by-FC and Ext-by-Conv. As stated in Section I, deep features extracted by Ext-by-Conv are yielded in receptive fields of a single size, while deep features obtained by Ext-by-FC can be viewed as being generated in receptive fields of different sizes. Hence, Ext-by-Conv performs worse than Ext-by-FC in classification accuracy, as shown in Section V(D). However, Ext-by-FC is a time-consuming method since the inference of the used CNN model is performed one time for each image patch. By contrast, Ext-by-Conv is much more efficient because the inference only needs to be executed one time to obtain all deep features.

To make BoDVW model have high feature extraction efficiency and high classification accuracy, we enhance deep features extracted by Ext-by-Conv at low added computation costs by supplementing the information obtained from receptive fields of different sizes. Considering that different convolutional layers have different receptive field sizes, we present a novel feature named ‘‘FD vector’’ that can roughly preserve the information of multiple deep features extracted by different convolutional layers. Each deep feature is enhanced by combining an FD vector to obtain a combined feature. The image representation vector of an image is generated using the combined features extracted from it.

In the following, we first illustrate how to generate FD vectors and how to combine deep features with FD vectors. Afterward, the reason why an FD vector can roughly preserve the information of multiple deep features is given. For clarity, Table 1 lists the main abbreviations used in this article and their meanings.

A. SOLUTION

1) FEATURE DIFFERENCE VECTOR

An FD vector records the differences among multiple deep features at a location. As shown in Figure 3, given a set $\{f^1, \dots, f^L\}$ where $f^l, l = 1, \dots, L$ denotes the deep feature at the location (i, j) on the output of the l -th used convolutional layer, then the FD vector d_{ij} is calculated as follows:

$$d_{ij} = \left[d_{1,2}^T, \dots, d_{k,l}^T, \dots, d_{L-1,L}^T \right]_{k < l, k=1, \dots, L, l=1, \dots, L}^T \in R^{[GL(L-1)/2] \times 1} \quad (1)$$

where,

$$d_{k,l} = \left[\|f_1^k - f_1^l\|_2, \dots, \|f_G^k - f_G^l\|_2 \right]^T$$

$$f^k = \left[(f_1^k)^T, \dots, (f_G^k)^T \right]^T, f^l = \left[(f_1^l)^T, \dots, (f_G^l)^T \right]^T \quad (2)$$

To attain more difference information, each D -dimensional feature vector is partition into G groups. Each group is a sub-vector with the same dimension D/G . The difference between two features f^k, f^l is described by a vector where the g -th element is the Euclidean distance between f_g^k and f_g^l . It is worth noting that the outputs of the convolutional layers used for obtaining FD vectors must have the same size.

The dimension of FD vectors is decided by G and L . Our experimental results show that when $L = 3$ and $G = 128$, the generated FD vectors are good enough to achieve high classification accuracy (shown in Section V(D)). In this case, the dimension of an FD vector is only 384. It is much lower than the sum of the dimensions of three deep features used to yield an FD vector, e.g., $2048 \times 3 = 6144$ when using

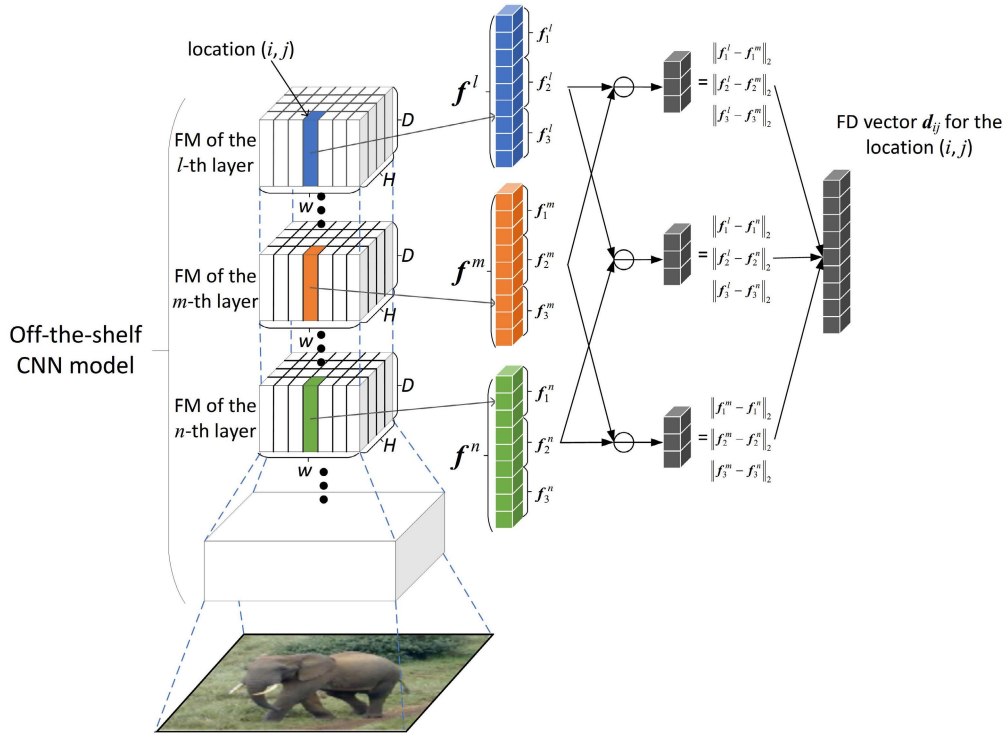


FIGURE 3. Toy example of generating a feature difference vector ($L = 3, G = 3$). FM: feature map.

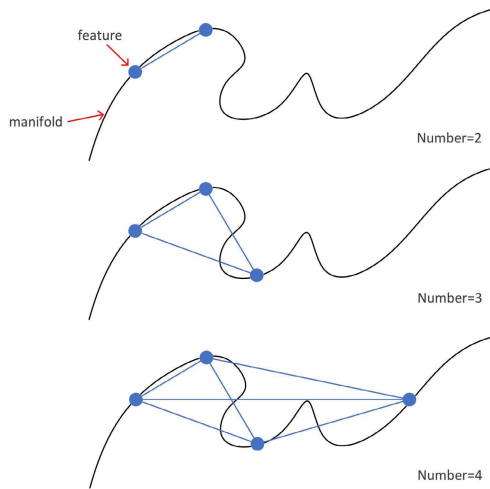


FIGURE 4. Illustration on the ability of FD vectors to roughly preserve the information of multiple deep features.

ResNet-50 as a feature extractor to extract 2048-dimensional deep features.

2) COMBINED FEATURE

Each of the deep features extracted at the used deepest-level convolutional layer is enhanced by combining an FD vector to obtain a combined feature. Specifically, given an FD vector $d_{i,j} \in R^{GL(L-1)/2 \times 1}$ and the deep features $[f_{i,j}^1, \dots, f_{i,j}^L] \in R^{D \times L}$ used to generate it, then the combined feature $u_{i,j}$ is formed by $u_{i,j} = [(f_{i,j}^L)^T, (\lambda d_{i,j})^T]^T \in R^{(GL(L-1)/2 + D) \times 1}$.

λ is a hyperparameter to balance $d_{i,j}$ and $f_{i,j}^L$, which can be decided by cross validation. The combined feature $u_{i,j}$ records not only the accurate information of $f_{i,j}^L$ but also the coarse information of $f_{i,j}^1, \dots, f_{i,j}^{L-1}$.

B. EXPLANATION

An FD vector can roughly preserve the information of multiple deep features used to generate it. The reason can be illustrated as follows.

Existing works have pointed out that high-dimensional features are approximately located on an irregular manifold. As shown in Figure 4, the features denoted by the blue circles are located on the manifold indicated by the black curve, and the length of a blue line denotes the Euclidean distance between two features. It is easily found that as the number N of blue circles increases, the potential locations that can yield the same distances as the ones among the blue circles become more and more scarce. The number P of the potential locations is infinite when $N = 2$, whereas P drops obviously when $N > 2$. Therefore, the Euclidean distances among multiple features ($N > 2$) possess the ability to preserve their information to some extent. With the increase of N , the ability to preserve their information will be improved. It is worth noting that if the features are too close, the improvement of the ability is limited.

Because the outputs of the convolutional layers used for generating FD vectors have the same size, all extracted deep features are in the same feature space. In other words, they can be seen as being located on the same manifold. Hence, for



FIGURE 5. Example images of 15-Scenes, TF-Flowers, and NWPU-RESISC45.

L deep features at a location, the distances among them can preserve their information to some extent. These distances can be recorded by an FD vector ($G = 1$), thus the FD vector naturally possesses the ability to preserve their information. To more finely describe the distance between two features, the distance vector calculated using their sub-vectors is recorded instead of a distance scalar. In fact, this practice can also be explained as calculating a special FD vector using $L \times G$ sub-vectors with the dimension D/G , which roughly preserves the information of these sub-vectors. The specialty is that for each sub-vector, only the distances from it to $L - 1$ different sub-vectors instead of all other sub-vectors are calculated. Furthermore, since the L deep features are yielded by different convolutional layers, they are not close in the feature space.

V. EXPERIMENTS

A. DATASETS

Our experiments are conducted on three datasets, i.e., 15-Scenes, TF-Flowers, and NWPU-RESISC45. As shown in Figure 5, they are taken from three scenarios, including scene classification, object recognition, and remote sensing image classification. The details on these datasets are listed below:

15-Scenes is a scene recognition dataset with 200 to 400 images per category. It consists of 4485 images spread over 15 categories such as bedroom, industrial, kitchen, living

room, and so on. 100 images are randomly taken from each category for training and the remaining ones for testing.

TF-Flowers is a challenging dataset including images from five different categories of flowers. These categories are Daisy, Dandelion, Tulips, Roses, and Sunflowers. Each category has about 600 to 900 images with different sizes and aspect ratios. For each category, 450 images and 150 images are randomly chosen for training and testing, respectively.

NWPU-RESISC45 is a remote sensing image classification dataset consisting of 31500 images divided into 45 scene categories. Each category has 700 images with the size of 256×256 pixels. The spatial resolution changes from about 30m to 0.2m per pixel. 140 images per category are used for training and 60 images per category for testing.

B. IMPLEMENTATION DETAILS

The implementation details of our method are as follows:

Feature Extraction: Two off-the-shelf CNN models, VGGNet-16 and ResNet-50 (implemented by PyTorch), are used in our experiments. Before extracting deep features, the two models are fine-tuned using all training samples (no data augmentation) for higher classification accuracy. In the fine-tuning process, the stochastic gradient descent algorithm (learning rate = 0.001, momentum = 0.9) is applied. The period of learning rate decay and the multiplicative factor are set to 8 and 0.1, respectively. For clarity, the suffix “(no-refined)” denotes the fine-tuned model of which only the fully-connected part is modified and re-trained. The suffix “(refined)” indicates the CNN model of which all layers are fine-tuned. The fully-connected part of ResNet-50 is the last fully-connected layer, and the fully-connected part of VGGNet-16 is the last three fully-connected layers. For VGGNet-16, the outputs of the last three convolutional layers (named “features.36”, “features.39” and “features.42” by Pytorch) are taken as the source of deep features, resulting in $3 \times 14 \times 14$ 512-dimensional deep features for each image. For ResNet-50, the outputs of the last three residual blocks (named “layer4.0.relu”, “layer4.1.relu” and “layer4.2.relu” by Pytorch) are taken as the source of deep features, leading to $3 \times 7 \times 7$ 2048-dimensional deep features for each image. FD vectors are generated by Formula (1) using extracted deep features, and combined features are yielded according to the method stated in Section IV(A). The hyperparameter λ is decided by cross-validation.

Dictionary Learning: The traditional clustering algorithm K -means is used to learn a visual dictionary based on combined features.

Feature Coding: As done in [3], LLC is adopted to encode combined features as coding vectors. The number of visual words for encoding each combined feature is set to 5.

Feature Pooling: The spatial pyramid [15] with the levels of 1×1 , 2×2 , and 4×4 is used to divide each image into 21 blocks. The coding vectors in each block are pooled together by maximum pooling. All pooling vectors are concatenated to form the image representation vector, followed

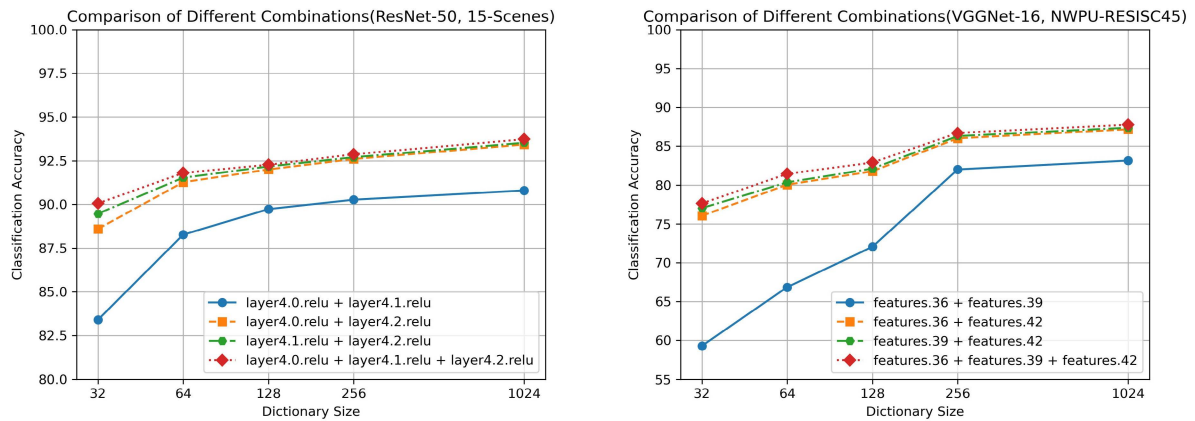


FIGURE 6. Comparison of Different Combinations on 15-Scenes and NWPU-RESISC45.

by $l_{1.5}$ -normalization (l_2 -normalizing the square roots of element values).

Classifier Training: A linear SVM is trained for each category in a one-versus-rest manner. The hyperparameter C is set to 2.

All experiments are performed 10 times to acquire reliable experimental results, and the average classification accuracy of 10 experiments is reported in this article. The accuracies obtained by Ext-by-FC are also reported for comparison. For Ext-by-FC, image patches of 96×96 pixels, 128×128 pixels, 160×160 pixels, 192×192 pixels with the step of 32 pixels are densely sampled from each image resized to the minimum side length of 224 pixels. The average pooling layer in ResNet-50 and the last fully-connected layer named “classifier.5” in VGGNet-16 are utilized to convert image patches to deep features. The settings on dictionary learning, feature coding, feature pooling, and classifier training are the same as stated above. All experiments are conducted on a computer with an Intel Core i9-10900 at $2.8\text{GHz} \times 10$ on 64GB RAM. The extraction of deep features and the fine-tune of CNN models are done with an NVIDIA GTX 2060(super) GPU for acceleration.

C. EVALUATION ON FEATURE DIFFERENCE VECTORS

In this subsection, we evaluate the classification performance of FD vectors under different parameter setups. The main parameters are the number L of convolutional layers used for generating FD vectors and the number G of groups (sub-vectors).

1) IMPACT OF THE COMBINATIONS OF DIFFERENT CONVOLUTIONAL LAYERS

The impact of the combinations of different convolutional layers on classification accuracy is investigated. The combinations of two or three layers are evaluated on 15-Scenes and NWPU-RESISC45 when VGGNet-16(refined) and ResNet-50(refined) are used as two feature extractors. Here,

the number G of the groups (sub-vectors) is set to 256 for ResNet-50 and 128 for VGGNet-16.

As shown in Figure 6, with the increase of the dictionary size, the classification accuracies of all combinations are improved. The combinations of three layers always achieve the highest classification accuracy under different dictionary sizes. This phenomenon implies that FD vectors generated using three layers are more discriminative than those obtained using two layers. Furthermore, the combinations of the lower two layers, i.e., the combination “layer4.0.relu + layer4.1.relu” and the combination “features.36 + features.39”, perform worse than other combinations in which the deepest convolutional layer (“layer4.2.relu”, “features.42”) is included. These results support the existing finding that a deeper layer can generate features with higher discriminability. In the following experiments, we use the combination of three layers to generate FD vectors.

2) IMPACT OF THE NUMBERS OF GROUPS

In this subsection, we elaborately evaluate the impact of the number G of groups (sub-vectors) on the classification accuracy. Here, the combination of three convolutional layers is applied to generate FD vectors, i.e., “features.36 + features.39 + features.42” of VGGNet-16 and “layer4.0.relu + layer4.1.relu + layer4.2.relu” of ResNet-50. The accuracies achieved with the dictionaries of different sizes (32, 64, 128, 256, 1024) are reported in Figure 7. For comparison, the accuracies of VGGNet-16(no-refined), VGGNet-16(refined), ResNet-50(no-refined), and ResNet-50(refined) are also reported.

As shown, as G increases, the classification accuracies obtained with the dictionaries of different sizes all improve significantly. This phenomenon implies that computing a distance vector between two deep features instead of a distance scalar is very useful. The larger G will lead to FD vectors with a higher dimension. For example, according to Formula (1), the dimension of FD vectors is only $(4 \times 3 \times 2)/2 = 12$ when

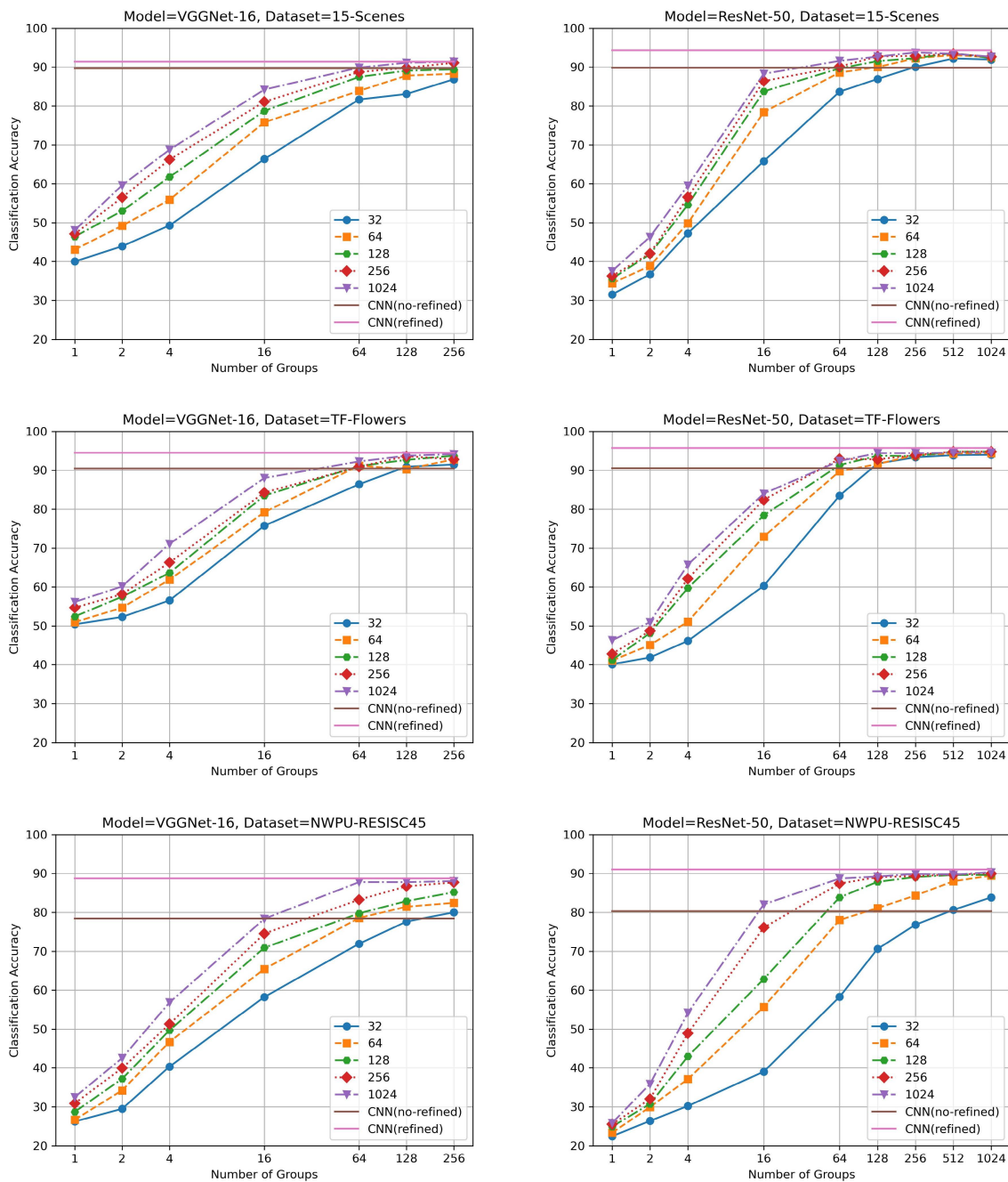


FIGURE 7. Classification accuracies of FD vectors generated under different parameter setups.

$G = 4$, while the dimension is increased to $(128 \times 3 \times 2)/2 = 384$ when $G = 128$. Furthermore, the practice of increasing G and the dictionary size can achieve higher accuracy. However, the accuracy gain brought by this practice is not obvious when G exceeds 64 or 128 and the dictionary size is 1024. For example, when the dictionary size is 1024, the accuracy gains obtained by increasing G from 128 to 256 on the three datasets are quite small. It is easily found from each sub-figure that the highest accuracy obtained using FD vectors yielded via the refined CNN model is superior to

the one obtained by the no-refined CNN model and very close to the one achieved by the refined CNN model.

D. EVALUATION ON COMBINED FEATURES

This subsection evaluates the effectiveness of using combined features to classify images. Here, if using VGGNet-16(refined) as a feature extractor, FD vectors yielded when $G = 128$ are combined with deep features, and if using ResNet-50(refined) as a feature extractor, FD vectors yielded when $G = 256$ are combined with

TABLE 2. Classification accuracies of different methods on 15-Scenes, TF-Flowers, and NWPU-RESISC45. The suffix “(Ext-by-FC)” denotes that deep features are generated by Ext-by-FC, and the suffix “(Ext-by-Conv)” indicates that deep features are generated by Ext-by-Conv.

	Feature	Extractor	15-Scenes(%)	TF-Flowers(%)	NWPU-RESISC45(%)
1	Deep Feature(Ext-by-Conv)	ResNet-50(refined)	94.37	95.81	90.52
2	FD vector	ResNet-50(refined)	93.73	94.8	89.96
3	Combined Feature	ResNet-50(refined)	94.59	96	90.98
4	Deep Feature(Ext-by-FC)	ResNet-50(refined)	94.8	96	91.6
5	Deep Feature(Ext-by-Conv)	VGGNet-16(refined)	92.86	94.7	89.03
6	FD vector	VGGNet-16(refined)	91.13	94.5	88.04
7	Combined Feature	VGGNet-16(refined)	93.2	94.89	89.19
8	Deep Feature(Ext-by-FC)	VGGNet-16(refined)	93.1	95.1	89.22
Other Methods					
9	ResNet-50(no-refined)		89.8	90.53	80.26
10	ResNet-50(refined)		94.27	95.73	91.03
11	VGGNet-16(no-refined)		89.67	90.44	78.43
12	VGGNet-16(refined)		91.4	94.53	88.7
13	Places365-GoogleNet [29]		91.25	-	-
14	Places365-VGGNet [29]		91.97	-	-
15	DSFL+CNN [30]		92.81	-	-
16	G-MS2F [31]		92.9	-	-
17	[32]		92.9	-	-
18	FTOTLM [33]		94.01	-	-
19	Khan et al. [4]		94.5	-	-
20	SDO + fc features [34]		95.88	-	-
21	DFF-ADML [26]		96.39	-	-
22	FTOTLM, data aug. [33]		97.4	-	-
23	[12]		-	88.16	-
24	TUNELoss [35]		-	90.43	-
25	[36]		-	95	-
26	VGGNet-16 + BoCF [11]		-	-	84.32
27	CNN-SC [37]		-	-	85.29
28	VGG_VD16 + SAFF [23]		-	-	87.86
29	ResNet-50 + EAM [38]		-	-	93.51
30	ResNet-101 + EAM [38]		-	-	94.29

TABLE 3. Average computation time spent on extracting features from an image.

	Feature	Extractor	Time on Feature Extraction (second)
1	Deep Feature(Ext-by-Conv)	ResNet-50(refined)	0.415s
2	Combined Feature	ResNet-50(refined)	0.455s
3	Deep Feature(Ext-by-FC)	ResNet-50(refined)	1.40s
4	Deep Feature(Ext-by-Conv)	VGGNet-16(refined)	0.1s
5	Combined Feature	VGGNet-16(refined)	0.136s
6	Deep Feature(Ext-by-FC)	VGGNet-16(refined)	1.16s

deep features. For comparison, the results obtained using only deep features to classify images are also listed. The dictionary size is chosen from 512, 1024, 2048, 4096 by cross-validation. For combined features, the hyperparameter λ is picked out from 0.001, 0.01, 0.1, 0.5, 1, 5, 10 by cross-validation.

Table 2 reports the classification accuracies of different methods. As shown in the 1st to the 8th rows, the practice of using combined features to classify images can result in higher accuracy than using only FD vectors and using only deep features extracted by Ext-by-Conv. Besides, this practice obtains comparable results to that of leveraging deep features extracted by Ext-by-FC to classify images. Especially when using VGGNet-16(refined) as a feature extractor, a higher accuracy of 93.2% is achieved using combined features. Table 3 lists the average computation time spent on extracting features from an image. As shown in the 1st, 3rd, 4th, 6th rows, Ext-by-FC takes more time

than Ext-by-Conv. When using VGGNet-16(refined) as a feature extractor, Ext-by-Conv is about 11.6 times faster than Ext-by-FC. Furthermore, as shown in the 1st, 2nd, 4th, 5th rows, generating combined features takes slightly more time than extracting deep features by Ext-by-Conv. However, the time spent on generating combined features is still much less than on extracting deep features by Ext-by-FC. In addition, since the structure of ResNet-50 is more complex than that of VGGNet-16, the computation time spent on feature extraction when using ResNet-50(refined) as a feature extractor is more than when using VGGNet-16(refined) as a feature extractor. Overall, it can be concluded that deep features extracted by Ext-by-Conv are enhanced at low added computation costs by combining FD vectors, resulting in comparable results to that of using deep features generated by Ext-by-FC to classify images. This conclusion indicates that the target of our work is achieved, i.e., enhancing deep features extracted by Ext-by-Conv to make BoDWW model

have high feature extraction efficiency and high classification accuracy.

Compared with directly applying fine-tuned CNN models to classify images (the 9th to the 12th rows), the BoDVW methods listed in the 1st to 8th rows perform better in classification accuracy. However, there is an exception that the accuracy of 90.98% obtained by using combined features on NWPU-RESISC45 is lower than 91.03% achieved by directly applying ResNet-50(refined). Furthermore, the classification performance of a BoDVW method strongly relies on the performance of the CNN model used as a feature extractor in the BoDVW method. As shown, since ResNet-50(refined) achieves higher accuracy than VGGNet-16(refined), the results (the 1st to the 4th rows) yielded using ResNet-50(refined) as a feature extractor are better than that (the 5th to the 8th rows) of using VGGNet-16(refined) as a feature extractor. These results imply that higher accuracy can be achieved when using a more advanced CNN model to extract features.

Compared with other methods listed in the 13th to the 30th rows, using combined features achieves the state-of-the-arts on TF-Flowers, but does not exceed SDO + fc features [34], DFF-ADML [26], and EAM [38] on 15-Scenes and NWPU-RESISC45. Since our method is just one for improving the BoDVW model, higher classification accuracy can be achieved using our method and other methods together (illustrated in Section VI).

VI. DISCUSSION

We propose a simple method for improving BoDVW model in this article. Our method can avoid the high computations costs of Ext-by-FC and achieve comparable results to Ext-by-FC. It can be extended from the following aspects.

1) The novel feature named “FD vector” can be combined with other features such as handcrafted features to solve other task-specific problems, such as medical image classification, remote sensing image classification, and so on.

2) Our method is just one for improving BoDVW model. It can be combined with other methods to achieve higher classification accuracy. For example, an advanced data augmentation scheme (e.g., [33]) is applied to enlarge training samples, and then an advanced CNN model (e.g., ResNeXt-50 [39]) is fine-tuned using enlarged training samples to extract deep features.

3) The idea to roughly preserve the information of multiple deep features in a low dimensional vector by computing the differences among them can be applied in the design of CNN architecture. For example, the attention module can be built based on the differences among the outputs of multiple convolutional layers.

VII. CONCLUSION

This article raises a simple method to make BoDVW model have high feature extraction efficiency and high classification accuracy. Our method enhances deep features extracted by Ext-by-Conv by supplementing the information obtained

from receptive fields of different sizes. Concretely, each deep feature is enhanced by combining an FD vector that can roughly preserve the information of multiple deep features extracted by different convolutional layers. Our experimental results on three public datasets show that our method can avoid the high computation costs of Ext-by-FC and achieve comparable results to Ext-by-FC. The future work we are pursuing is to solve task-specific classification problems by combining the novel feature “FD vector” with other features such as handcrafted features.

REFERENCES

- [1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Sep. 2014, pp. 1–12.
- [2] A. Bosch, A. Zisserman, and X. Muñoz, “Scene classification using a hybrid generative/discriminative approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.
- [3] Z. Jie and S. Yan, “Robust scene classification with cross-level LLC coding on CNN features,” in *Proc. Asian Conf. Comput. Vis.*, in Lecture Notes in Computer Science, Apr. 2015, pp. 376–390.
- [4] S. H. Khan, M. Hayat, M. Bennamoun, R. Togneri, and F. A. Sohel, “A discriminative representation of convolutional features for indoor scene recognition,” *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3372–3383, Jul. 2016.
- [5] Y. Huang, Z. Wu, L. Wang, and T. Tan, “Feature coding in image classification: A comprehensive study,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–506, Mar. 2014.
- [6] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
- [7] The TensorFlow Team. *Flowers*. Accessed: Aug. 22, 2021. [Online]. Available: http://download.tensorflow.org/example_images/flower_photos.tgz
- [8] J. P. Cohenm, P. Morrison, and L. Dal. *COVID-19 Image Data Collection*. Accessed: Jul. 17, 2021. [Online]. Available: <https://github.com/ieee8023/covid-chestxray-dataset>
- [9] G.-S. Xie, X.-Y. Zhang, S. Yan, and C.-L. Liu, “Hybrid CNN and dictionary-based models for scene recognition and domain adaptation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1263–1274, Jun. 2017.
- [10] A. Diba, A. M. Pazandeh, and L. Van Gool, “Deep visual words: Improved Fisher vector for image classification,” in *Proc. Int. Conf. Mach. Vis. Appl.*, May 2017, pp. 186–189.
- [11] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, “Remote sensing image scene classification using bag of convolutional features,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1735–1739, Oct. 2017.
- [12] M. Saini and S. Susan, “Bag-of-visual-words codebook generation using deep features for effective classification of imbalanced multi-class image datasets,” *Multimedia Tools Appl.*, vol. 80, no. 14, pp. 20821–20847, Mar. 2021.
- [13] C. Sitaula and S. Aryal, “New bag of deep visual words based features to classify chest X-ray images for COVID-19 diagnosis,” 2020, *arXiv:2012.15413*.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [15] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Oct. 2006, pp. 2169–2178.
- [16] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.
- [17] Y. Huang, Z. Wu, L. Wang, and C. Song, “Multiple spatial pooling for visual object recognition,” *Neurocomputing*, vol. 129, pp. 225–231, Apr. 2014.
- [18] F. Sadeghi and M. F. Tappen, “Latent pyramidal regions for recognizing scenes,” in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2012, pp. 228–241.
- [19] Y. Jia, C. Huang, and T. Darrell, “Beyond spatial pyramids: Receptive field learning for pooled image features,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3370–3377.

- [20] M. Dammak, M. Mejdoub, and C. B. Amar, "Histogram of dense sub-graphs for image representation," in *IET Image Process.*, vol. 9, no. 3, pp. 184–191, Mar. 2015.
- [21] N. Antropova, B. Q. Huynh, and M. L. Giger, "A deep feature fusion methodology for breast cancer diagnosis demonstrated on three imaging modality datasets," *Med. Phys.*, vol. 44, no. 10, pp. 5162–5171, Jul. 2017.
- [22] C. Wang, A. Elazab, J. Wu, and Q. Hu, "Lung nodule classification using deep feature fusion in chest radiography," *Comput. Med. Imag. Graph.*, vol. 57, pp. 10–18, Apr. 2017.
- [23] R. Cao, L. Fang, T. Lu, and N. He, "Self-attention-based deep feature fusion for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 43–47, Jan. 2021.
- [24] H. Wang and Y. Yu, "Deep feature fusion for high-resolution aerial scene classification," *Neural Process. Lett.*, vol. 51, no. 1, pp. 853–865, Sep. 2019.
- [25] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 183–186, Feb. 2018.
- [26] C. Wang, G. Peng, and B. De Baets, "Deep feature fusion through adaptive discriminative metric learning for scene recognition," *Inf. Fusion*, vol. 63, pp. 1–12, Nov. 2020.
- [27] S. Mei, K. Yan, M. Ma, X. Chen, S. Zhang, and Q. Du, "Remote sensing scene classification using sparse representation-based framework with deep feature fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 5867–5878, May 2021.
- [28] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.
- [29] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1454–1464, Jul. 2018.
- [30] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang, "Learning discriminative and shareable features for scene classification," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 552–568.
- [31] P. Tang, H. Wang, and S. Kwong, "G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition," *Neurocomputing*, vol. 225, pp. 188–197, Feb. 2017.
- [32] D. Giveki, "Scale-space multi-view bag of words for scene categorization," *Multimedia Tools Appl.*, vol. 80, no. 1, pp. 1223–1245, Jan. 2021.
- [33] S. Liu, G. Tian, and Y. Xu, "A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter," *Neurocomputing*, vol. 338, pp. 191–206, Apr. 2019.
- [34] X. Cheng, J. Lu, J. Feng, B. Yuan, and J. Zhou, "Scene recognition with objectness," *Pattern Recognit.*, vol. 74, pp. 474–487, Feb. 2018.
- [35] M. Streeter, "Learning effective loss functions efficiently," 2019, *arXiv:1907.00103*.
- [36] S. Giraddi, S. Seeri, P. S. Hiremath, and G. N. Jayalaxmi, "Flower classification using deep learning models," in *Proc. Int. Conf. Smart Technol. Comput., Electr. Electron. (ICSTCEE)*, Oct. 2020, pp. 130–133.
- [37] A. Qayyum, A. Malik, N. M. Saad, and M. Mazher, "Designing deep CNN models based on sparse coding for aerial imagery: A deep-features reduction approach," *Eur. J. Remote Sens.*, vol. 52, no. 1, pp. 221–239, Mar. 2019.
- [38] Z. Zhao, J. Li, Z. Luo, J. Li, and C. Chen, "Remote sensing image scene classification based on an enhanced attention module," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1926–1930, Nov. 2021.
- [39] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.



XIANGSHI WANG received the master's degree in mathematics and computer science from Nanjing Normal University, China, in 2004. She is currently a Lecturer in technology of Internet of Things with the Wuxi Institute of Technology, China. She has worked on machine vision and its application technology. Her research interests include machine vision and pattern recognition.

