FEATURE SPARSITY IN CONVOLUTIONAL NEURAL NETWORKS FOR SCENE CLASSIFICATION OF REMOTE SENSING IMAGE

Wei Huang[1], Qi Wang[1], Xuelong Li[1]*

[1]School of Computer Science and Center for OPTical IMagery Analysis and Learning(OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China.

ABSTRACT

Recently, the analysis of remote sensing images has attracted a lot of attention. In the domain of scene classification, deep learning methods, especially convolutional networks (CNNs), currently achieve the best results. Although the classification performance has reached a high level, there are still some factors limiting the improvement of classification accuracy. Based on obeservation of remote sensing scene images, we fing that some scenes are quite similar though they belong to different classes. To improve the classification performance between different scenes with similar characteristics, we propose a significant Feature Sparsity Layer that can be esaily embedded into various convolutional network architectures. The proposed layer can inhibit the confusing features meanwhile stress the discriminative features, and it is used to sparse the multi-layer feature map, which is extracted by the convolutional layers. The proposed method achieves the state-of-the-art results on three datasets UC Merced Land Use, Aerial Image Data and OPTIMAL-31, and competitive result on dataset WHU-RS19.

***Index Terms*—** Remote sensing image, scence classification, CNNs, feature sparsity

1. INTRODUCTION

Benefited from the rapid development of remote sensing equipments, many researches on remote sensing images have been significantly explored, including scene classification, disaster detection, hyperspectral image classification [1] and so on. Compared with normal RGB images, remote sensing images have its characteristic bucause of its capture mode. Usually they cover a large area containing abundant spatial information with a overhead view. Scene classification of remote sensing images is a basic but challenging task because of various classes and complex spatial information, and researchers have proposed many different methods to improve the performance of scene classification. According to the ways of extracting features, these methods can be roughly divided into two categories as following:

(a) Airport_1 (b) Airport_2 (c) Bridge_1 (d) Bridge_2

Fig. 1. Airport and Bridge belong to different scenes but their features are similar such as long stride road and dense green vegetation.

A. Traditional Methods. This type of methods are first used for scene classification and they are based on handcrafted features including global features and local features. Global features, *e. g.* color histograms and texture descriptors, can be directly delivered into the classifier, meanwhile local features, such as scale-invariany feature transform, need to be firstly integrated to an entire representation before being sent into the classifier. What's more, these individual features can be fused to generate more comprehensive features with some certain methods. However, this type of features only contain low-level spatial information with the lack of hign-level semantic information.

B.Deep Learning Methods. With the rapid development of hardware resources and huge labeled datasets, deep learning methods, especially convolutional neural networks (CNNs) such as [2] [3] [4] [5], have achieved the best results in the domain of scene classification of remote sensing images. A typical disadvantage of CNNs is that there are abundant parameters that need to be optimized, which can lead to the overfitting problem. Fortunately, the CNN architectures can be pre-trained on huge image database ImageNet before being applied in scene classification. And the main advantage of CNNs is that the neural networks can automatically learn the discriminative features for scene classification. Compared with handcrafted features, the deep learning features contain not only low-level spatial information, but also high-level semantic information.

*Corresponding Author.

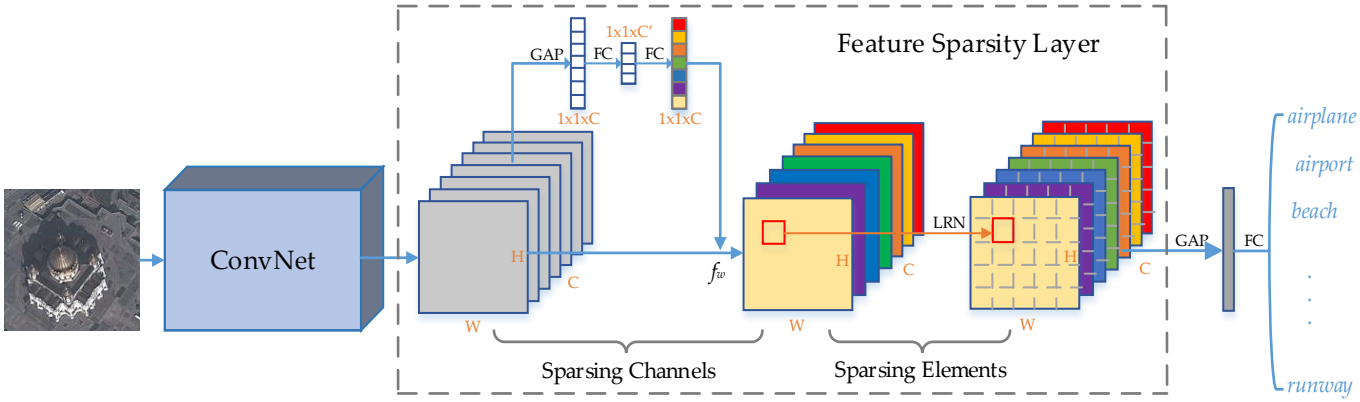

Fig. 2. The whole classification framework with the proposed feature sparsity layer. "GAP", "FC" and "LRN" represent global average pooling, fully-connected layer and local response normarlization, respectively. And f_w is the function that weights the features according to the corrsponding channel they are in.

Despite the performance of scene classification is already excellent, there are some problems need to be solved. Based on obeservation of remote sensing scene images, we find that some scenes are quite similar though they belong to different classes. As shown in Fig. 1, there are two scenes of Airport and Bridge. Both of them have long stride road and dense green vegetation, which are quite similar and confusing features. Meanwhile, there are also some discriminative features like terminal buildings in Airport and the river in Bridge.

To address this problem, we propose a Feature Sparsity Layer (FSL) motivated by Squeeze-and-Excitation Networks (SEnet) [6] and Local Response Normalization (LRN) [7]. SEnet is firstly proposed to weight different features in units of channels, but it can also be regarded as a channel-wise sparsity operation. And LRN is used to inhibit the features at the same position in the adjacent channels inspired by "lateral inhibitio" in Neurobiology, such that it can be seen as an element-wise sparsity operation. The proposed FSL combines them reasonably with the effectiveness of stressing the discrimanative features and inhibiting the confusing features. Similarly, [8] proposes a sparse operation for features. The contributions of our work can be summarized as follows:

1. We propose a significant Feature Sparsity Layer that weights high-level semantic features with respect to channels and elements. The sparsed high-level semantic features are more effective for scene classification of remote sensing images.

2. The Feature Sparsity can be easily embedded into various CNN architectures and and help them to breakthrough their upper limits of scene classification.

3. CNN architecture Resnet101 attatched by the proposed Feature Sparsity Layer achieves excellent results on several public scene datasets of remote sensing images, including the state-of-the-art results on three datasets UC Merced Land Use, Aerial Image Data and OPTIMAL-31, and the competitive result on dataset WHU-RS19.

2. METHOD

The whole classification framework with the proposed feature sparsity layer is illustrated in Fig. 2. The proposed FSL only plays an role of sparsing features, without changing the dimension of feature map. FSL can be composed of the following two parts: 1) Sparsing channels. 2) Sparsing elements.

2.1. Sparsing Channels

In convolutional networks, each channel of feature map can be regarded as an individual unit at a view of semantic information. However, different channels should have the unequal contribution to scene classification as mentioned above. Thus it is necessary to assign different channels with different weights to achieve the purpose of sparsing channels. Refering to [6], we adopt the following steps:

Step A: Calculating the global average value of each channel by using Global Average Pooling (GAP). The input is the multi-layer feature map U, which is extracted by the convolutional layers of CNN architectures. Its size is H x W x C, which denote width, height and the number of channels, respectively. Element at position (i, j) in c-th channel is denoted as $u_c(i, j)$. And the output is a vector of 1 x 1 x C, denoted as $w \in \mathbb{R}^c$. So the c-th element of w is calculated by:

$$w(c) = \frac{1}{H \times W} \sum_{i=0}^{H} \sum_{j=0}^{W} u_c(i, j) \tag{1}$$

Step B: Capturing the dependencies between channels. To fulfill this objective, nolinear operations (two fully-connected layers with activation funciton of Sigmod that are not shown

in Fig. 2.) are added after Global Average Pooling. The first fully-connected layer reduces the dimension of w from C to C/16, meanwhile the secend fully-connected layer restores the dimension from C/16 to C. The output of this step is denoted as w' and it is calculated by:

$$w' = \sigma(f_2(\sigma(f_1(w))) \tag{2}$$

where f_1 and f_2 represent the two fully-connected layer, and σ is the activation function Sigmod. $Step\ C$: Assigning features the corresponding weight according to the channels which they are in. Input of this step is multi-layer feature map U and the weight vector w', and the output is the weighted multi-layer feature map U' with the same size as U. The element at position $(i,\ j)$ of c-th channel of U' is calculated by:

$$u'_c(i,j) = w'(c) * u_c(i,j) \tag{3}$$

2.2. Sparsing Elements

After finishing the operation of channel-wise sparsity, it is necessary to perform the element-wise sparsing operation. We make use of Local Response Normalization (LRN) to fulfill this objective. Referring to [7], wo apply LRN in the following format:

$$u''_c(i,j) = u'_c(i,j)/(k + \alpha \sum_{c=max(0,c-n/2)}^{min(C-1,c+n/2)} (u'_c(i,j))^2)^\beta \tag{4}$$

where $u_c{}'(i,\ j)$ is the element of input U', meanwhile $u_c{}''(i,\ j)$ is the element of output U'' which is the feature map sparsed by LRN. C is the number of the channels. k, α, β and n are the hyper parameters, and in this paper their value are 0.0001, 0.75, 1 and 2, respectively. After finishing these two sparsing operations, the sparsed multi-layer feature map U'' is used for classification.

3. EXPRIMENTS

To prove the effectiveness of the proposed Feature Sparsity Layer, we embed it to several classic CNN architectures and choose the architecture with the best perfermance. Then we train and test the architecture with FSL on several public scene datasets of remote sensing images.

3.1. Datasets

We experiment on four public scene datasets of remote sensing images: UC Merced Land-Use Data Set (UCM), WHU-RS19 (WHU), Aerial Image Data Set (AID) and OPTIMAL-31 Data Set (OPD). Because these datasets have no standard criteria of spliting (no benchmarks), we randomly split each of them into training set and test set at a splitting ratio that is consistent with other related literature. Their basic information in experiments is summarized in Table 1.

Table 1. BASIC INFORMATION OF THE DATASETS ON EXPERIMENTS.

	WHU	UCM	OPD	AID
classes	19	21	31	30
size	600 x 600	256 x 256	256 x 256	600 x 600
training images	597(60%)	1680(80%)	1488(80%)	5000(50%)
test images	408(40%)	420(20%)	372(20%)	5000(50%)

3.2. Training Strategy

During the training process, we select Stochastic Gradient Descent (SGD) to optimize the whole model with the following parameters: The momentum is 0.9, and the learning rate is set to 0.0001 without weight decay. The batch size in experiments is 64. We use TenCrop to augment training set: all training samples are resized to 256 x 256, then an area of 224 x 224 is randomly cropped from five corners (upper left, lower left, upper right, lower right and the center) and five flipped corners. When validating and testing, images are resized to 256 x 256 and the center area of 224 x 224 are used to evaluate the models. To fairly compare the original CNN architectures and the architectures attached by FSL, it is the condition for stopping training the models that the accuracy no longer increases for 50 epochs.

3.3. Results

Table 2. EXPERIMENTS ON AID OF FOUR CNN ARCHITECTURE WITHOUT/WITH FEATURE SPARSITY LAYER.

Architecture	Accuracy
Bninception	86.14%
Bninception-FSL	**87.08%**
InceptionResnetv2	93.74%
InceptionResnetv2-FSL	**94.44%**
Resnet34	94.00%
Resnet34-FSL	**94.06%**
Resnet101	95.10%
Resnet101-FSL	**95.88%**

Firstly it is necessary to find an optimal combination of the CNN architecture and FSL. We add the Feature Sparsity Layer to four CNN architectures including Bninception [10], InceptionResnetv2 [11], Resnet34 and Resnet101 [12] (All of them are pre-trained on ImageNet). In this step, dataset AID id used to evaluate their performance and the experimental results are shown in Table 2. From the result we can find FSL can help all of the CNN architectures to breakthrough their upper limits with the increase of 0.64% on average. And performance of resnet101 with FSL is the best combination among all the models, so we choose resnet101 with FSL as the constract model in the next experiments.

Table 3. COMPARISON RESULTS ON FOUR PUBLIC DATASETS OF THE PROPOSED METHOD AND THE OTHER MODELS.

Model	WHU	UCM
Resnet101-FSL	98.77%	**99.52%**
Resnet101	98.53%	98.81%
ARCNet-VGG16 [5]	**99.75±0.25%**	99.12±0.40%
Combing Scenarios I and II [3]	98.89%	98.49%
Fusion by Addition [9]	98.65±0.43%	97.42±1.79%
VGG-VD-16 [4]	96.05±0.91%	95.21±1.20%
Model	OPD	AID
Resnet101-FSL	**95.16%**	**95.88%**
Resnet101	94.62%	95.10%
ARCNet-VGG16 [5]	92.70±0.35%	93.10±0.55%
Combing Scenarios I and II [3]	—	—
Fusion by Addition [9]	—	91.87±0.36%
VGG-VD-16 [4]	89.12±0.35%	89.64±0.36%

Then we experiment resnet101 with FSL on the other three datasets and compare all the results in this experiments with the existing state-of-the-art results. They are summarized in Table 3. From the table we can see that resnet101 with the proposed Feature Sparsity Layer achieves the excellent results: the best classification accuracy on three datasets including UC Merced Land Use, Aerial Image Data and OPTIMAL-31, and a competitive performance on dataset WHU-RS19. Because the dataset WHU-RS19 is randomly splited which leads to the ratio of training samples is less than 60% in fact (see Table 1), the accuracy of resnet101 with FSL is not as good as ARCNet-VGG16. These results prove the effectiveness of the proposed Feature Sparsity Layer.

4. CONCLUSION

In this paper, we find that there are some quite similar semantic features between some certain scenes through observation of the remote sensing images. Based on this observation, we propose Feature Sparsity layer, which can be easily embedded into various CNN architectures, to stress the discriminative features and inhibit the confusing features. Experimental results prove the effectiveness of the proposed method.

5. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant U1864204 and 61773316, Natural Science Foundation of Shaanxi Province under Grant 2018KJXX-024, and Project of Special Zone for National Defense Science and Technology Innovation.

6. REFERENCES

[1] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," 2018.

[2] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, pp. 230–241, Jan. 2018.

[3] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, pp. 14680–14707, 2015.

[4] G.-S. et al Xia, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 3965–3981, July 2017.

[5] 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*, pp. 1–13, Sep. 2018.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] J. Yu, Y. R, and D. Tao, "Click prediction for web image reranking using multimodal sparse coding," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2019–2032, 2014.

[9] 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, pp. 4775–4784, May 2017.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017, vol. 4.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.