Chapter

Utilization of Deep Convolutional Neural Networks for Remote Sensing Scenes Classification

Chang Luo, Hanqiao Huang, Yong Wang and Shiqiang Wang

Abstract

Deep convolutional neural networks (CNNs) have been widely used to obtain high-level representation in various computer vision tasks. However, for the task of remote scene classification, there are no sufficient images to train a very deep CNN from scratch. Instead, transferring successful pre-trained deep CNNs to remote sensing tasks provides an effective solution. Firstly, from the viewpoint of generalization power, we try to find whether deep CNNs need to be deep when applied for remote scene classification. Then, the pre-trained deep CNNs with fixed parameters are transferred for remote scene classification, which solve the problem of time-consuming and parameters over-fitting at the same time. With five well-known pre-trained deep CNNs, experimental results on three independent remote sensing datasets demonstrate that transferred deep CNNs can achieve state-of-the-art results in unsupervised setting. This chapter also provides baseline for applying deep CNNs to other remote sensing tasks.

Keywords: convolutional neural network, remote sensing, scene classification, deep learning, generalization power

1. Introduction

Remote sensing image processing achieves great advances in recent years, from low-level tasks, such as segmentation, to high-level ones, such as classification. [1–7] However, the task becomes incrementally more difficult as the level of abstraction increases, going from pixels, to objects, and then scenes. Classifying remote scenes according to a set of semantic categories is a very challenging problem, because of high intra-class variability and low interclass distance. [5–9] Therefore, the more representative and higher-level representations are desirable and will certainly play a dominant role in scene-level tasks. The deep convolutional neural network (CNN), which is acknowledged as the most successful and widely used deep learning model, attempts to learn high-level features corresponding to high level of abstraction [10]. Its recent impressive results for classification and detection tasks bring dramatic improvements beyond the state-of-the-art records on a number of benchmarks [11–14]. In theory, considering the subtle differences among categories in remote scene classification, we may attempt to form high-level representations for remote scenes from CNN activations. However, the acquisition of large-scale well-annotated remote sensing datasets is costly, and it is easy to

over-fit when we try to train a high-powered deep CNN with small datasets in practice [15]. In other words, with limited remote sensing dataset, deep CNNs work perfectly on the training data but do not generalize well to test data, resulting in poor performance eventually.

ImageNet[1] is a large-scale dataset, which offers a very comprehensive database of more than 1.2 million categorized natural images of 1000+ classes [16]. Deep CNN models trained upon this dataset serve as the backbone for many segmentation, detection, and classification tasks on other datasets. Moreover, some very recent works have demonstrated that the representations learned from deep CNNs pre-trained on large datasets such as ImageNet can be transferable to image classification task [17]. Some works also start to apply them to remote sensing field and obtain state-of-the-art results for some specific datasets [15, 18, 19]. However, the generalization power of features learned from deep CNNs fades evidently when the features of remote sensing images become different with that of natural images in the ImageNet dataset [15, 18]. Therefore, to solve the problem discussed above, the generalization power of deep CNNs plays the key role. We find that the generalization power of a deep CNN is relative to its depth. A deeper architecture trained by large-scale dataset may lead to a more general hypothesis for remote scenes. To our surprise, features learned from deeper layers are more general than that learned from shallower layers in a deep CNN when we transfer them for remote scene classification. This overturns the traditional view that features in shallow layers of a deep CNN are composed of basic visual patterns (e.g., salient edges and borders) and they are more general for test data. Inspired by this, we evaluate the generalization power of transferred deep CNN for remote scenes in different conditions and explore the proper way to apply deep CNNs to remote scene classification with limited remote sensing data.

We conduct extensive experiments with transferred deep CNN and evaluate the generalization power of it on different remote sensing datasets that vary in space information. The results show that the depth of CNNs contributes to the generalization power of them. Features from deeper layers are more general for test data and brings better performance in remote scene classification. Then, we conduct extensive experiments with different pre-trained deep CNNs such as CaffeNet [13], GoogLeNet [20], and ResNet [21]. This chapter hardly contains any deep or new techniques, and our study so far is mainly empirical. However, a thorough report on generalization power of deep CNNs for remote scene classification has tremendous value for applying deep CNNs to remote sensing images. A satisfied answer to this question would not only help to make features of remote scenes more interpretable in deep CNNs, but it might also lead to more principled and reliable deep architecture design. Our main contributions are summarized as follows:

1. We thoroughly investigate how transferred deep CNNs work for remote scene classification with limited remote sensing data and how the generalization power of them affect their performance.

2. This chapter challenges the classical view of features learned in deep CNNs by showing that high-level features learned in deeper layers are more general than basic features (e.g., salient edges and borders) learned in shallower layers. Features learned in shallow layers of deep CNNs are not general enough for remote scenes. This leads us to believe that depth of CNNs enhances the

[1] http://www.image-net.org/challenges/LSVRC/

generalization power of the learned features and it is essential for remote scene classification.

3. Based on various pre-trained deep CNNs, we evaluate our proposed method on different remote sensing datasets that vary in space and spectrum. The results show that our proposed method can learn better features for remote scenes. With "unsupervised settings," our proposed method achieves state-of-the-art performance on some public remote scene datasets.

The rest of the chapter is organized as follows. Section 2 presents successful pre-trained deep CNNs nowadays and the way to transfer them for remote scene classification. Section 3 analyzes the generalization power of features in different layers of transferred deep CNN. Experiments are presented in Section 4, and we conclude the chapter in Section 5.

2. Transferred deep CNNs for remote scene classification

Convolutional neural networks are generally presented as systems of interconnected processing units which can compute values from inputs leading to an output that may be used on further units. The typical architecture of a deep CNN is composed of multiple cascaded layers with various types.

Among the different types of layers, the convolutional one is the responsible for capturing the features from the images. The first layers usually obtain low-level features (like edges, lines, and corners), while the others get high-level features (like structures, objects, and shapes). The process made in this type of layer can be decomposed into two phases: (i) the convolution step, where a fixed-size window runs over the image, with some stride, defining a region of interest and (ii) the processing step that uses the pixels inside each window as input for the neurons that, finally, perform the feature extraction from the region. The continuous form and discrete form of convolutional operation can be expressed as Eqs. (1) and (2), respectively:

$$h(x,y) = i*k(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(u,v)k(x-u,y-v)dudv \tag{1}$$

$$H(x,y) = I*K(x,y) = \sum_m \sum_n I(m,n)K(x-m,y-n) \tag{2}$$

As to the input map, the convolutional operation can be further illustrated by **Figure 1**:

Conventionally, a nonlinear function is provided after the convolutional operation, which is usually called activation function. There are a lot of alternatives for activation function, such as sigmoid function $\frac{1}{1+e^{-x}}$ and tanh function $\frac{e^x-e^{-x}}{e^x+e^{-x}}$. The most popular activation function nowadays is called rectified linear unit (ReLU). ReLU has several advantages when compared to others: (i) works better to avoid saturation during the learning process, (ii) induces the sparsity in the hidden units, and (iii) does not face gradient vanishing problem as with sigmoid and tanh function. The mathematic form of the ReLU can be shown as follows:

$$a = \begin{cases} z, & if\ z>0 \\ 0, & otherwise \end{cases} \quad \Leftrightarrow \quad a = f(z) = \max(0,z) \tag{3}$$

Typically, after obtaining the convolved feature activations, we would next like to aggregate statistics of these features at various locations, and this aggregation

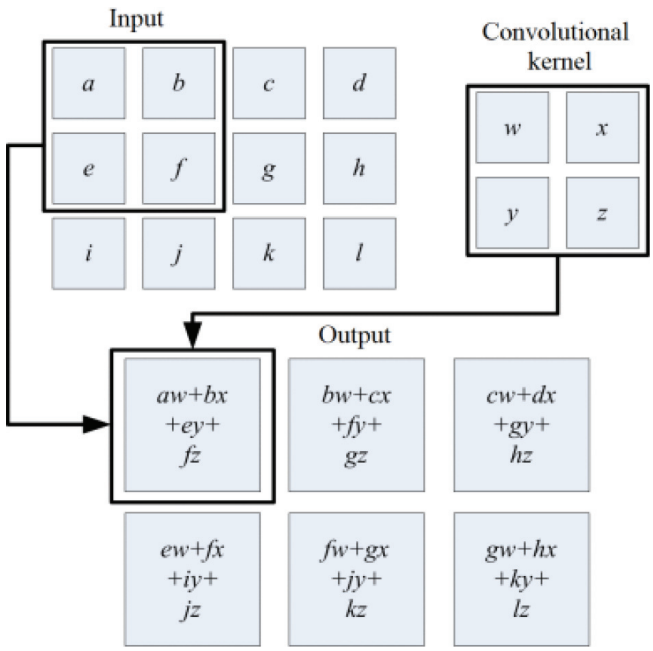

Figure 1.
Convolutional operation.

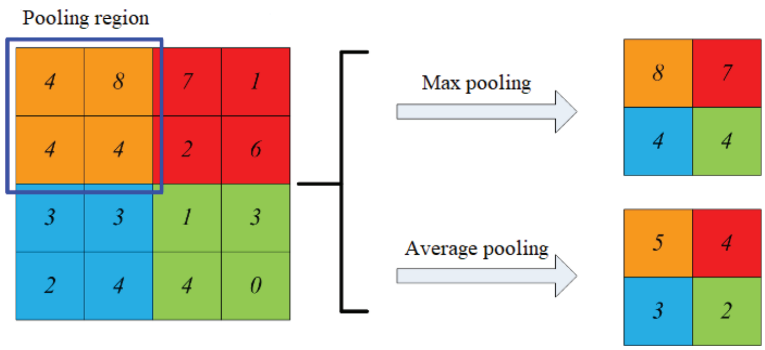

Figure 2.
Pooling operation.

operation is called pooling operation. Pooling operation within the pooling region translates convolved feature activations into pooled features, which are much lower in dimension and can improve classification results (i.e., less over-fitting). Pooling regions are usually contiguous areas in the convolved feature maps, and the pooled features are usually generated from the same filter. Then these pooled features would be "translation invariant." Although several novel pooling approaches have been proposed, max pooling and average pooling are still the most commonly used approaches as shown in **Figure 2**.

After several convolutional and pooling layers, there are the fully connected ones, which take all neurons in the previous layer and connect them to every single neuron in its layer. Since a fully connected layer occupies most of the parameters, over-fitting can easily happen. To prevent this, the dropout method was employed as shown in **Figure 3**. This technique randomly drops several neuron outputs, which do not contribute to the forward pass and backpropagation anymore. This neuron drops are equivalent to decreasing the number of neurons of the network,

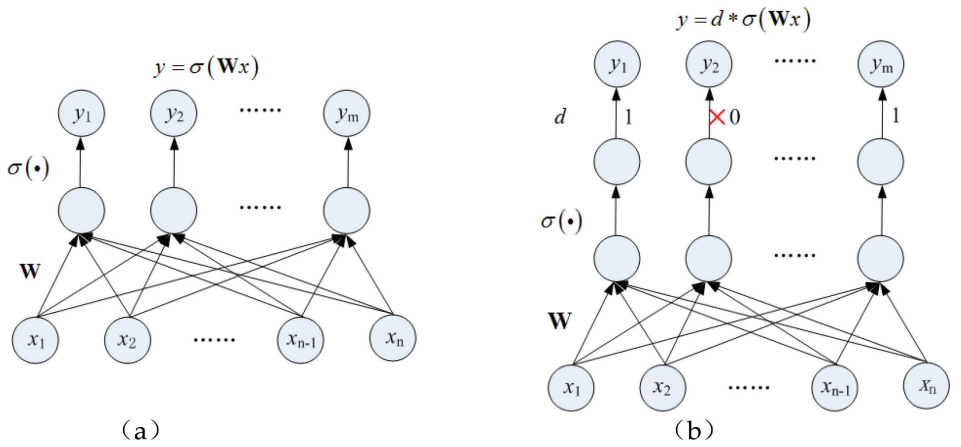

Figure 3.
Dropout method. (a) No dropout and (b) dropout.

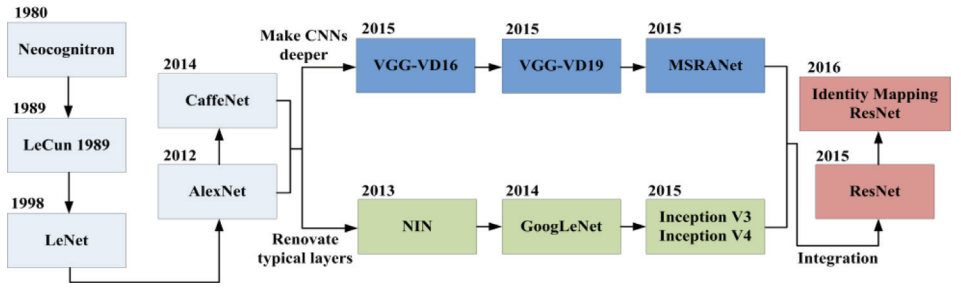

Figure 4.
Evolution of the structure of deep CNNs.

improving the speed of training, and making model combination practical, even for deep networks.

Finally, after all the convolution, pooling, and fully connected layers, a classifier layer may be used to calculate the class probability of each instance.

Based on the typical architecture of deep CNN, AlexNet [11], CaffeNet [13], VGG-VD [14], MSRA-Net [22], NIN [23], GoogLeNet [20], Inception V3 [24], Inception V4 [25], and ResNet [21] all proved to be effective in detection or classification tasks and achieve state-of-the-art performance.

In summary, we demonstrate the evolution of deep CNNs' structure in **Figure 4**:

However, these successful deep CNNs discussed above do not achieve good performance as we expected, when we directly apply them for remote scene classification. An effective solution, recently explored in [15, 18, 20], is to transfer deep features trained on ImageNet dataset to remote sensing images. Deep CNNs pretrained by ImageNet dataset can be treated as fixed feature extractors. In a feedforward way, they extract global feature representation of the remote sensing images. With the global representation, a simple classifier can implement remote scene classification. Taking a step further, fine-tuning strategy is usually used for deeper layers of transferred deep CNNs to further improve the performance of them for remote scene classification. Typically, the first few layers are frozen, because low-level features can better fit remote scenes, and deeper layers are allowed to keep learning by training them with remote sensing images. Taking AlexNet, for example, we show the fine-tuning strategy in **Figure 5**.

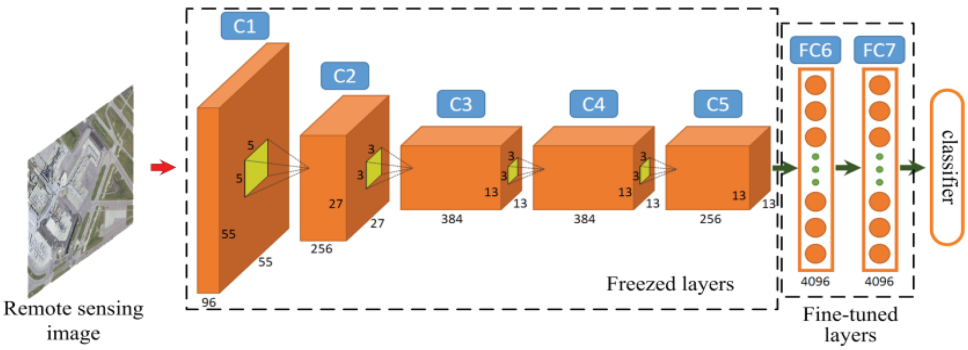

Figure 5.
Strategy of fine-tuning deeper layers of AlexNet.

Although the strategy of fine-tuning deeper layers of transferred deep CNNs with remote sensing images achieves near-perfect performance in remote scene classification [18], we challenge the theory basis of this strategy by showing that not all low-level features in shallow layers are general enough for remote scenes; some of them even shows very poor generalization power in transferring process. We find that the depth of transferred CNNs enhances the generalization power of them and guarantees a general hypothesis for remote scene classification. The detailed results are discussed in Section 3. This find in transferred deep CNNs gives an answer to the very recent discussion about whether generalization power of deep CNNs comes from sheer memorization or available hypothesis.

3. Generalization power of features in different layers of transferred deep CNN

As mentioned in Section 2, when transferring deep CNNs pre-trained by ImageNet for remote scene classification, we typically assume that features (e.g., salient edges and borders) in the shallow layers are generic, while features in the deep layers are more specific to the dataset used for pre-training and thus need to be fine-tuned by the target dataset. Therefore, the traditional strategy of transferring pre-trained deep CNNs for remote scene classification is to freeze the shallow layers and fine-tune the last deep layers. However, this assumption drives us to the question that how the "depth" of transferred deep CNNs affect the features of remote scenes in the transferring process. To answer this question, we take CaffeNet pre-trained by ImageNet, for example, and thoroughly analyze features of remote scenes in different layers of it when we transfer it for remote scene classification on UC Merced dataset[2].

Firstly, we take a close look into features of remote sensing image in the first convolutional layer of the pre-trained CaffeNet. In **Figure 6**, we visualize the convolutional filters of the first convolutional layer. These convolutional filters are learned by pre-training the CaffeNet with ImageNet dataset. We can see that the former filters are learned for extracting edges in different directions and the later filters are learned for extracting different colors. For example, the first, fifth, and ninth filters are mainly used to extract features in the right lower oblique direction,

[2] http://vision.ucmerced.edu/datasets/landuse.html

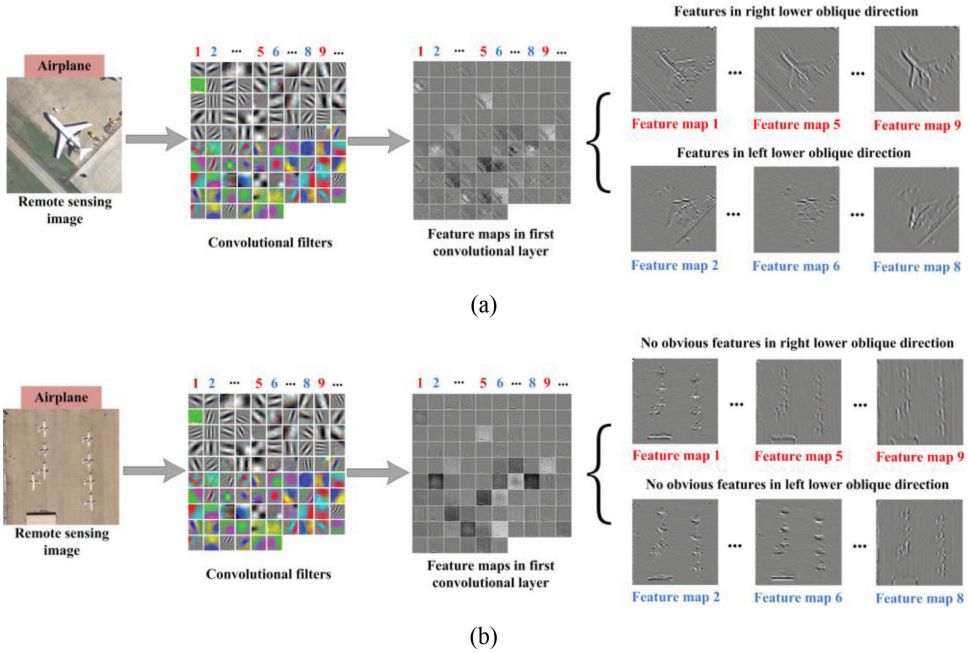

Figure 6.
Feature maps of (a) one remote sensing image and (b) another one within the same remote scene class extracted by convolutional filters in the first layer of pre-trained CaffeNet.

while the second, sixth, and eighth filters are mainly used to extract features in the left lower oblique direction. Based on the architecture of pre-trained CaffeNet, we can obtain 96 feature maps in the first convolutional layer by applying these convolutional filters to remote sensing image. In **Figure 6(a)**, we find that the first, fifth, and ninth feature maps contain features of the input image in the right lower oblique direction, while the second, sixth, and eighth feature maps contain features of the input image in the left lower oblique direction. However, in **Figure 6(b)**, we cannot see obvious features in these two directions in the corresponding feature maps. The input images in **Figure 6(a)** and **(b)** belong to the same remote scene class. However, features of them extracted by filters in the first convolutional layer of pre-trained CaffeNet are very different from each other. Compared with daily optical images in ImageNet dataset, remote sensing images are much more sophisticated. Some convolutional filters in shallow layers of pre-trained CaffeNet may be effective for some remote sensing image while affecting little about some other remote sensing images. Not all features in shallow layers of pre-trained CaffeNet are general for remote sensing images.

Furthermore, we try to visualize features of the input remote sensing image learned in deeper layers of the pre-trained CaffeNet. However, as we can see in **Figure 7**, feature maps of the remote sensing image become increasingly fuzzy from the second convolutional layer. With the increase of depth, representations of remote scene become more and more abstract. In order to reveal how the depth of pre-trained CaffeNet affects the generalization power of features in it, we intuitively reflect the distribution of features learned from the two input remote sensing images in **Figure 8** by using the t-SNE algorithm. [26, 27] In **Figure 8**, we use the t-SNE algorithm to visualize feature maps in different convolutional layers by giving each datapoint a location in a 2-D map.

Figure 8 shows the separability of features learned in different convolutional layers of pre-trained CaffeNet when we apply it on two different remote sensing

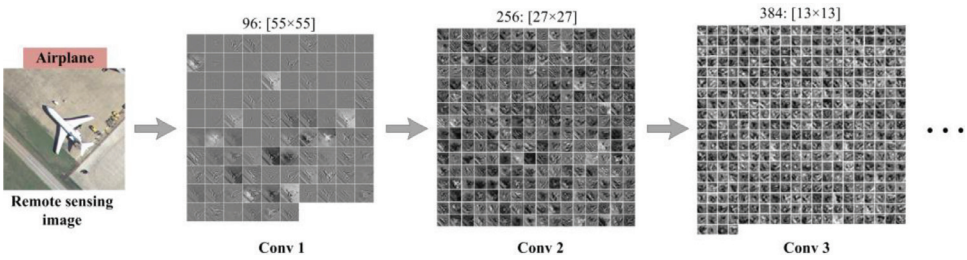

Figure 7.
Feature maps of the remote sensing image in different layers of pre-trained CaffeNet.

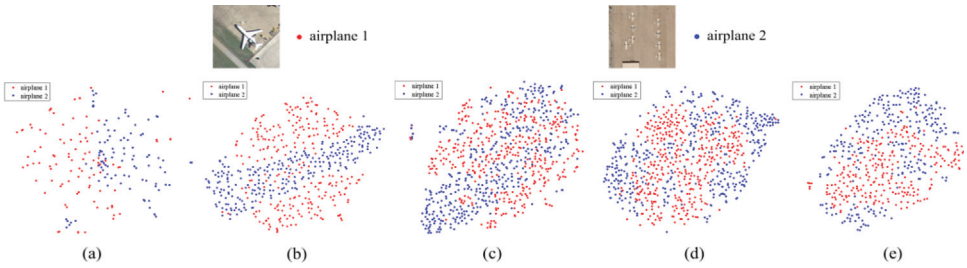

Figure 8
2-D visualization of feature maps in (a) the first convolutional layer, (b) the second convolutional layer, (c) the third convolutional layer, (d) the fourth convolutional layer, and (e) the fifth convolutional layer of pre-trained CaffeNet. The t-SNE algorithm proposed in [26, 27] is used to visualize the high-dimensional representations.

images that belong to the same remote scene class. In **Figure 8(a)**, the 2-D features of the two input images are separated to each other obviously in the first convolutional layer. Notably, from **Figure 8(a)–(e)**, the deeper the layer, the more overlap between features of the two remote sensing images we can observe. Therefore, in contrast to common belief that features in shallow layer are more generic, they are susceptible to changes in input remote sensing images. Indeed, filters of the first convolutional layer are similar to HOG, SURF, or SIFT (edge detectors, color detectors, texture, etc.). They give representative information for different input images. However, this information also conveys the specific characteristics of the dataset used to pre-train the CaffeNet. As a result, features extracted in shallow layers of pre-trained CaffeNet may be not general enough for remote scene classification in the transferring process. On the other hand, it seems that the depth of pre-trained CaffeNet enhances the generalization power of features in it. Regardless of the specific meaning of edges or colors, high-level features in deeper layer represent the sematic meaning of the input remote sensing image. Based on this analysis of features in pre-trained CaffeNet, we believe that depth of pre-trained CNNs brings general hypothesis for remote scene classification. It plays an important role when we apply pre-trained CNNs to the task of remote scene classification.

4. Experiments

The main objective of this chapter is to evaluate different deep CNNs transferred for remote scene classification. Therefore, we organize the experiments for transferred deep CNNs with various deep CNN architectures and various remote sensing datasets. We try to explore the answer for the problem where the generalization power comes from in deep CNNs and find the proper way to apply deep

CNNs for remote scene classification. All the developed codes rely on the MatConvNet[3] framework which provides a complete deep learning toolkit for training and testing models. In addition, it should be noted that all the experiments are performed on HP z820 with two Intel (R) Xeon (R) CPUs with 2.60 GHz of clock and 32GB of RAM memory. NVIDIA Quadro K2000 series is used as graphic processing units.

4.1 Experimental setup

In this section, we carry out a number of experiments based on different architectures of deep CNNs. To evaluate the effectiveness of pre-trained deep CNNs transferred for remote scene classification, we conduct experiments on three remote sensing datasets. These three datasets are different in spatial and spectral information. We compare the performance of pre-trained deep CNNs with the state-of-the-art results in these three datasets. We must note that except learning the classifier, all the experiments are unsupervised.

The three publicly available datasets used in our experiments are as follows:

UC merced land use dataset. This dataset is composed of 2100 overhead scene images divided into 21 land use scene classes. Each class consists of 100 aerial images measuring 256 × 256 pixels, with a spatial resolution of 0.3 m per pixel in the red-green-blue color space. The example images for each class are shown in **Figure 9**. This dataset was extracted from aerial orthoimagery downloaded from the United States Geological Survey (USGS) National Map of the following US regions: Birmingham, Boston, Buffalo, Columbus, Dallas, Harrisburg, Houston, Jacksonville, Las Vegas, Los Angeles, Miami, Napa, New York, Reno, San Diego, Santa Barbara, Seattle, Tampa, Tucson, and Ventura. So far, this dataset is the most popular and has been widely used for the task of remote scene classification and retrieval. [28]

WHU-RS dataset[4]. Collected from Google Earth, this dataset is composed of 950 aerial scene images with 600 × 600 pixels, which are uniformly distributed in 19

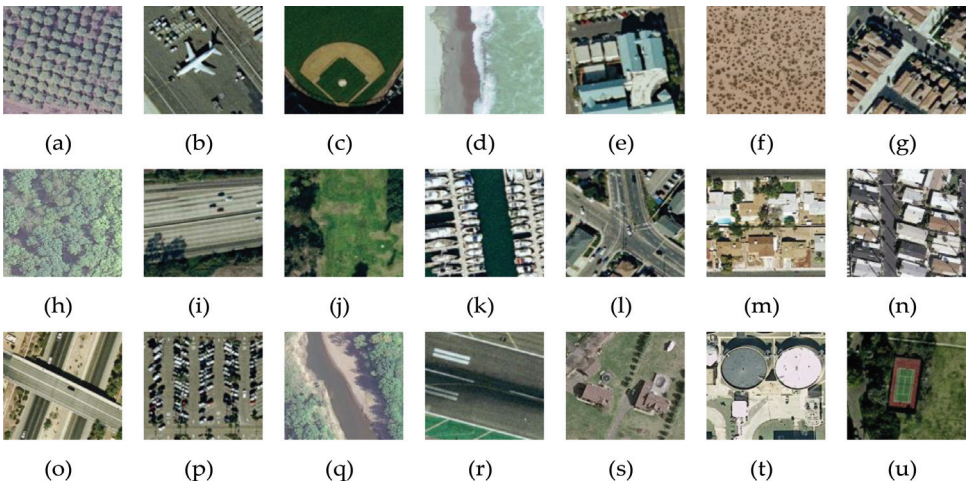

Figure 9.
One example image for each class of the UC Merced land use dataset. (a) Agricultural; (b) airplane; (c) baseball diamond; (d) beach; (e) buildings; (f) chaparral; (g) dense residential; (h) forest; (i) freeway; (j) golf course; (k) harbor; (l) intersection; (m) medium residential; (n) mobile home park; (o) overpass; (p) parking lot; (q) river; (r) runway; (s) sparse residential; (t) storage tanks; (u) tennis court.

[3] http://www.vlfeat.org/matconvnet/

[4] http://www.tsi.enst.fr/~xia/satellite_image_project.html.

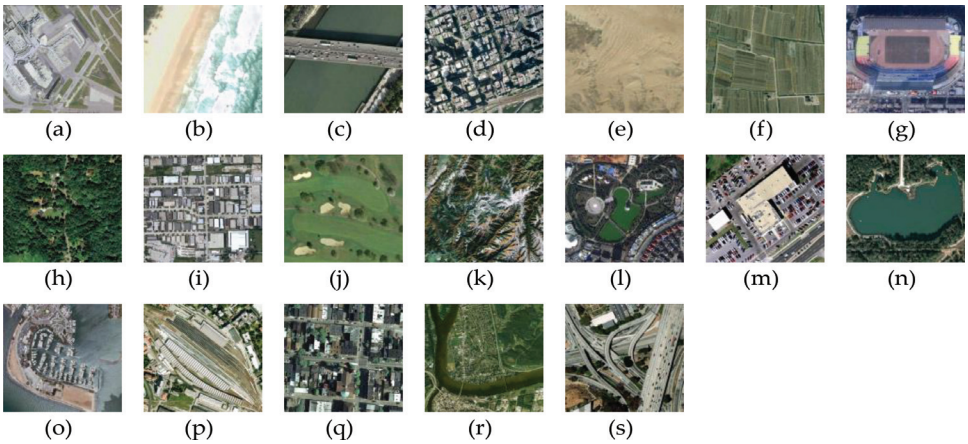

Figure 10.
One example image for each class of the WHU-RS dataset. (a) Airport; (b) beach; (c) bridge; (d) commercial; (e) desert; (f) farmland; (g) football field; (h) forest; (i) industrial; (j) meadow; (k) mountain; (l) park; (m) parking lot; (n) pond; (o) port; (p) railway; (q) residential; (r) river; (s) viaduct.

scene classes, 50 for each class. With spatial resolution up to 0.5 m and spectral bands of red, green, and blue, the example images for each class are shown in **Figure 10**. This dataset is challenging due to the high variations in resolution, scale, orientation, and illuminations of the images.

Brazilian coffee scenes dataset[5]. This dataset consists of only two scene classes (coffee class and non-coffee class), and each class has 1438 image tiles with a size of 64 × 64 pixels cropped from SPOT satellite images over four counties in the State of Minas Gerais, Brazil: Arceburgo, Guaranesia, Guaxupe, and Monte Santo. This dataset considered the green, red, and near-infrared bands because they are the most useful and representative ones for distinguishing vegetation areas. **Figure 11** shows three example images for each of the coffee and non-coffee classes in false colors.

In the experiments, we divide all the datasets in fivefolds. For UC Merced dataset, WHU-RS dataset, and Brazilian coffee scenes dataset, each of the five folds contains 420 images, 190 images, and 600 images, respectively. Then, the classification accuracy and standard deviation are calculated with fivefold cross-validation. Five well-known pre-trained deep CNNs (AlexNet [11], CaffeNet [13], VGG-VD16 [14], GoogLeNet [20], and ResNet [21]) descripted in Section 2 are used to test the effectiveness of pre-trained deep CNNs in the experiments. As we analyzed before, all the experiments are in unsupervised framework except learning the classifier.

4.2 Experiment results of remote scene classification

We evaluate transferred deep CNNs for the task of remote scene classification based on the five well-known deep CNN architectures (AlexNet, CaffeNet, VGG-VD16, GoogLeNet, and ResNet) pre-trained by ImageNet. For the strategy of transferring deep CNNs for remote scene classification, we use the five pre-trained deep CNNs to extract high-level features from input images. These input images are resized to 227 × 227 for pre-trained AlexNet and CaffeNet and 224 × 224 for pre-trained VGG-VD16, GoogLeNet, and ResNet by down-sampling or up-sampling operation. Linear SVM is used as classifier.

[5] www.patreo.dcc.ufmg.br/downloads/brazilian-coffee-dataset/

Figure 11.
Example images of the Brazilian coffee scene dataset in false colors. (a)–(c) coffee class; (d)–(f) non-coffee class.

Pre-trained deep CNN	UC merced		WHU-RS		Brazilian coffee scenes	
	Ac (%)	SD	Ac (%)	SD	Ac (%)	SD
AlexNet	94.51	0.94	94.57	0.61	85.14	1.26
CaffeNet	94.12	1.05	94.67	0.75	84.97	1.54
VGG-VD16	94.43	0.68	94.76	0.72	84.12	0.97
GoogLeNet	94.57	0.98	94.68	1.01	84.06	1.16
ResNet-50	74.14	5.89	75.12	5.36	60.54	7.22
ResNet-101	72.36	5.96	72.85	5.09	59.39	6.68
ResNet-152	72.48	4.35	72.81	4.42	59.62	6.81

Table 1
Remote scene classification results of the five well-known pre-trained deep CNNs on three different remote sensing datasets.

With various pre-trained deep CNN models and remote sensing datasets, the remote scene classification performances are shown in **Table 1**. In **Table 1**, Ac and SD denote accuracy and standard deviation, respectively.

In the experiment, pre-trained deep CNNs are directly used as feature extractors in an unsupervised manner. By removing the last fully connected layer, the rest parts of pre-trained deep CNNs extract high-dimensional feature vectors of remote sensing images. These feature vectors are considered as final image representation followed by a linear SVM classifier. From **Table 1**, we can see that all transferred deep CNNs generated from AlexNet, CaffeNet, VGG-VD16, and GoogLeNet achieve state-of-the-art performance. Pre-trained deep CNNs show strong generalization power in the transferring process. In addition to our surprise, the most successful deep CNNs to date, ResNets fail to obtain a good experiment result, no matter their layers are 50, 101, or 152. In ResNets, shortcut connections bring less parameters and make the network much easier to optimize. At the same time, the direct connection between input and output brings poor generalization ability when we transfer them for other tasks. On the other hand, as shown in **Figure 11**, the spatial information of remote sensing images in the Brazilian coffee scene dataset is very simple. However, these remote sensing images are not optical (green-red-infrared). In **Table 1**, the relatively poor performance on this dataset comes from the difference in spectral information when we are transferring pre-trained deep CNNs for remote scene classification.

In order to test the performance of transferred deep CNNs for each remote scene class, in **Figure 12**, we draw the confusion matric of the experiment results on UC Merced dataset based on pre-trained CaffeNet.

In **Figure 12**, the experiment results in perfect or near-perfect accuracy for most of the scene categories. The relatively lower classification accuracy lies in the categories of building, dense residential, medium residential, and tennis court.

However, all these classes have some very "close" neighbors. Taking dense residential as example, it suffers the presence of very close classes, like buildings and medium residential, which we cannot even distinguish by eyes. Taking pre-trained CaffeNet, for example, **Figure 13** shows the detail changes of an optical remote sensing image.

Abbreviated as "conv" and "fc," reconstructions of convolutional feature maps in the former network layers and that of fully connected layers are shown in **Figure 13**. **Figure 13** shows that the representations of convolutional layers are still photographically similar with the remote sensing image to some extent, although they become fuzzier and fuzzier from "conv1" to "conv5." In addition, the fully connected layers rearrange the information from lower layers to generate representations that are more abstract. They compose of parts (e.g., the wings of airplanes) similar but not identical to the ones found in the original image.

In **Table 2**, we compare our best result achieved via transferred deep CNNs with various state-of-the-art methods on the UC Merced dataset. With a straightforward and simple framework, transferred deep CNN achieves outstanding performance on this dataset. We must note that our proposed method just provides basic

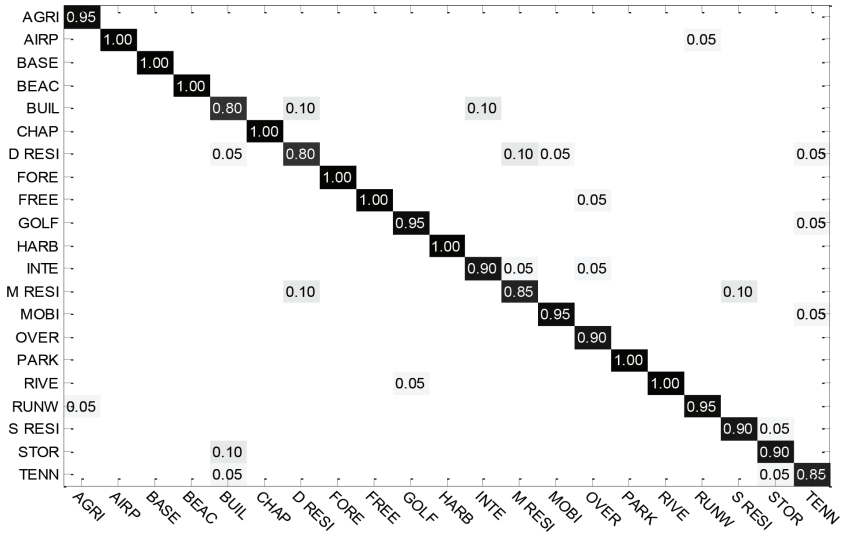

Figure 12.
Confusion matrices of classification accuracies on UC Merced dataset based on pre-trained CaffeNet.

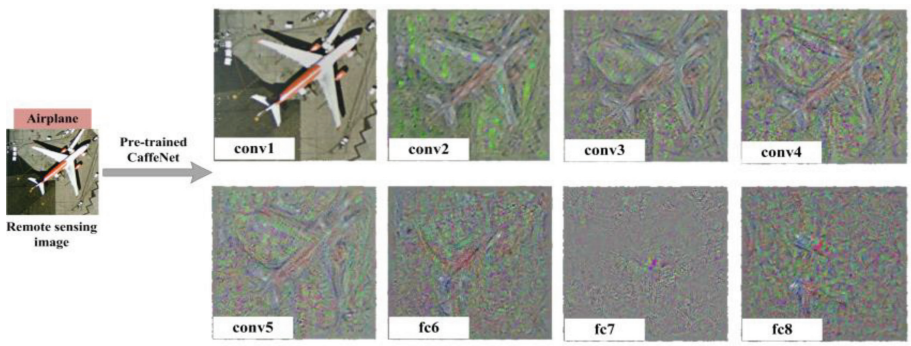

Figure 13.
Reconstruction of deep CNN activations from different layers of transferred CaffeNet. The method presented in [29] is used for visualization.

Method	Reference	Accuracy (%)
SCK	[28]	72.52
SPCK++	[30]	77.38
BRSP	[31]	77.80
UFL	[5]	81.67
CCM-BOVW	[32]	86.64
mCENTRIST	[33]	89.90
MSIFT	[34]	90.97
COPD	[35]	91.33
Dirichlet	[36]	92.80
VLAT	[10]	94.30
MCMI-based	[37]	88.20
PSR	[38]	89.10
UFL-SC	[39]	90.26
Partlets	[40]	91.33
Sparselets	[41]	91.46
FBC	[42]	85.53
LPCNN	[43]	89.90
MTJSLRC	[44]	91.07
SSBFC	[45]	91.67
CTS	[46]	93.08
Transferred GoogLeNet	—	**94.57**

Table 2
Classification accuracy (%) of reference and transferred deep CNN on the UC Merced dataset.

framework to directly transfer pre-trained deep CNNs for remote scene classification in an unsupervised manner. The effectiveness of fine-tuning approach is much dependent on the amount of images in remote sensing dataset, and the computation time of it is more demanding compared with our proposed strategy [15].

5. Discussion

To solve the problem that deep CNNs tend to over-fit when trained with limited remote sensing dataset, generalization power of deep CNNs plays the key role. In this chapter, we try to transfer deep CNNs pre-trained by daily images for remote scene classification and provide an insight for the generalization power of features in the transferred deep CNNs. From the extensive experiments above, the deep architecture of CNNs, which extracts semantic features of remote scenes, has been proven to be critical for remote scene classification. Specifically, several practical observations from the experiments and some limitations of our study are summarized as follows:

From **Table 1**, we can see that with our proposed method the classification accuracies of UC Merced dataset and WHU-RS dataset can both achieve state-of-the-art results which are near 95%. In addition, small standard deviation of

classification accuracy suggests that our proposed method is stable when applied for remote scene classification. To our surprise, the most successful deep CNNs to date, ResNets, fail to obtain good experiment result when we transfer it for remote scene classification, no matter their layers are 50, 101, or 152. Shortcut connections in ResNets bring poor generalization ability when we transfer them to remote scenes. [21] This phenomenon indicates that not all successful deep CNNs are suitable for transferring to the task of remote scene classification.

Different from the traditional view that all basic features (e.g., salient edges and borders) in shallow layers of a deep CNN are more general than that learned in deep layers, we find some features in shallow layer of deep CNNs show poor generalization power when we transfer them for remote scene classification. High-level features learned in deeper layers of transferred deep CNNs are more general than these basic features.

In the remote sensing field, the scale of remote sensing datasets will be larger and larger. On the other hand, the structure of deep CNN will be optimized, and the parameters in it will be less and less. [47] Therefore, we could get more and more useful information from remote sensing datasets, which provide a priori knowledge for pre-trained deep CNNs and result in better generalization power.

Based on our study, the future research directions of applying deep CNNs for remote scene classification may be as follows. Firstly, as we discussed above, when transferring the most successful ResNet for remote scene classification, it does not work as we expected. What is the proper architecture of deep CNN that is suitable to transfer to remote scenes? Secondly, instead of directly transferring pre-trained deep CNNs for remote scene classification, could we replace some basic features that show poor generalization power in shallow layers of transferred deep CNN? Finally, with more and more remote sensing information coming into our sight, how can we use these a priori knowledge when we apply deep CNNs for remote scene classification?

6. Conclusion

In this chapter, we have presented a framework to investigate the effectiveness of transferred deep CNNs for remote scene classification. We test transferred deep CNNs for different remote sensing datasets and take a close look into the generalization power of features in them.

The two main conclusions of this work are that (1) without shortcut connections in the deep architecture as ResNet dose, most CNNs transferred from well-known pre-trained deep CNNs achieve state-of-the-art performance in remote scene classification. (2) We further confirm the conclusion in the background of remote scene classification that the generalization power derived from deep architectures brings general hypothesis. Compared with basic features (e.g., salient edges and borders), features in deeper layers are more general for remote scenes. Experiments on three remote sensing datasets with different image resolutions have provided insightful information. Transferred deep CNN improves the classification accuracy of remote scenes on UC Merced dataset with a gain up to 1.49% compared with other methods. High-level feathers in deeper layers of transferred deep CNNs are more general for remote scene classification and result in satisfied performance in unsupervised setting.

We believe our work in this chapter provides a thorough analysis about the generalization power of transferred deep CNNs for remote scene classification. It can serve as a good baseline for people to apply deep CNNs to other remote sensing datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61601499 and No. 61601505. All the funds above can cover the costs to publish in open access.

Conflict of interest

The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Author details

Chang Luo[1*†], Hanqiao Huang[2†], Yong Wang[1] and Shiqiang Wang[3]

1 Troops of 78092, Cheng Du, China

2 Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, China

3 Air and Missile Defense College, Air Force Engineering University, Xi'an, China

*Address all correspondence to: luochang1988@126.com

† These authors contributed equally to this work and should be considered co-first authors.

References

[1] Wang J, Qin Q, Li Z, et al. Deep hierarchical representation and segmentation of high resolution remote sensing images. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE; 2015. pp. 4320-4323

[2] Nijim M, Chennuboyina RD, Al AW. A supervised learning data mining approach for object recognition and classification in high resolution satellite data. International Journal of Computer, Electrical, Automation, Control and Information Engineering. 2015;**9**(12):2319-2323

[3] Vakalopoulou M, Karantzalos K, Komodakis N, et al. Building detection in very high resolution multispectral data with deep learning features. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE; 2015. pp. 1873-1876

[4] Zhou W, Shao Z, Diao C, et al. High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder. Remote Sensing Letters. 2015; **6**(10):775-783

[5] Cheriyadat AM. Unsupervised feature learning for aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing. 2014; **52**(1):439-451

[6] Xu Y, Huang B. Spatial and temporal classification of synthetic satellite imagery: Land cover mapping and accuracy validation. Geo-spatial Information Science. 2014;**17**(1):1-7

[7] Yang W, Yin X, Xia GS. Learning high-level features for satellite image classification with limited labeled samples. IEEE Transactions on Geoscience and Remote Sensing. 2015; **53**(8):4472-4482

[8] Shao W, Yang W, Xia GS. Extreme value theory-based calibration for the fusion of multiple features in high-resolution satellite scene classification. International Journal of Remote Sensing. 2013;**34**(23):8588-8602

[9] Romero A, Gatta C, Camps-Valls G. Unsupervised deep feature extraction for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing. 2016;**54**(3):1349-1362

[10] Negrel R, Picard D, Gosselin PH. Evaluation of second-order visual features for land-use classification. In: 2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI). IEEE; 2014. pp. 1-5

[11] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. 2012. pp. 1097-1105

[12] Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229; 2013

[13] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia. ACM; 2014. pp. 675-678

[14] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556; 2014

[15] Castelluccio M, Poggi G, Sansone C, et al. Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv: 1508.00092; 2015

[16] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical

image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE; 2009. pp. 248-255

[17] Nanni L, Ghidoni S. How could a subcellular image, or a painting by van Gogh, be similar to a great white shark or to a pizza? Pattern Recognition Letters. 2017;**85**:1-7

[18] Penatti OAB, Nogueira K, dos Santos JA. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015. pp. 44-51

[19] Hu F, Xia G-S, Hu J, et al. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sensing. 2015;**7**:14680-14707

[20] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 1-9

[21] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385; 2015

[22] He K, Sun J. Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 5353-5360

[23] Lin M, Chen Q, Yan S. Network in network. arXiv preprint arXiv: 1312.4400; 2013

[24] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567; 2015

[25] Szegedy C, Ioffe S, Vanhoucke V. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv: 1602.07261; 2016

[26] Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. 2008;**9**(Nov): 2579-2605

[27] Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. Journal of Machine Learning Research. 2014;**15**(1):3221-3245

[28] Yang Y, Newsam S. 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. ACM; 2010. pp. 270-279

[29] Mahendran A, Vedaldi A. Understanding deep image representations by inverting them. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2015. pp. 5188-5196

[30] Yang Y, Newsam S. Spatial pyramid co-occurrence for image classification. In: 2011 International Conference on Computer Vision. IEEE; 2011. pp. 1465-1472

[31] Jiang Y, Yuan J, Yu G. Randomized spatial partition for scene recognition. In: Computer Vision–ECCV 2012. Berlin, Heidelberg: Springer; 2012. pp. 730-743

[32] Zhao LJ, Tang P, Huo LZ. Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model. Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2014;**7**(12):4620-4631

[33] Xiao Y, Wu J, Yuan J. mCENTRIST: A multi-channel feature generation mechanism for scene categorization. IEEE Transactions on Image Processing. 2014;**23**(2):823-836

[34] Avramović A, Risojević V. Block-based semantic classification of high-resolution multispectral aerial images. Signal, Image and Video Processing. 2016;**10**(1):75-84

[35] Cheng G, Han J, Zhou P, et al. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. ISPRS Journal of Photogrammetry and Remote Sensing. 2014;**98**:119-132

[36] Kobayashi T. Dirichlet-based histogram feature transform for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. pp. 3278-3285

[37] Ren J, Jiang X, Yuan J. Learning LBP structure by maximizing the conditional mutual information. Pattern Recognition. 2015;**48**(10):3180-3190

[38] Chen S, Tian YL. Pyramid of spatial relations for scene-level land use classification. IEEE Transactions on Geoscience and Remote Sensing. 2015; **53**(4):1947-1957

[39] Hu F, Xia GS, Wang Z, et al. Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification. Journal of Selected Topics in Applied Earth Observations and Remote Sensing. IEEE; 2015;**8**(5):13

[40] Cheng G, Han J, Guo L, et al. Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images. IEEE Transactions on Geoscience and Remote Sensing. 2015; **53**(8):4238-4249

[41] Cheng G, Han J, Guo L, et al. Learning coarse-to-fine sparselets for efficient object detection and scene classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 1173-1181

[42] Hu F, Xia GS, Hu J, et al. Fast binary coding for the scene classification of high-resolution remote sensing imagery. Remote Sensing. 2016;**8**(7):555

[43] Zhong Y, Fei F, Zhang L. Large patch convolutional neural networks for the scene classification of high spatial resolution imagery. Journal of Applied Remote Sensing. 2016;**10**(2): 025006-025006

[44] Kunlun Qi, Wenxuan Liu, Chao Yang, et al. High resolution satellite image classification using multi-task joint sparse and low-rank representation. Preprints, 7 November 2016, doi:10.20944/preprints201611. 0036.v1, (www.preprints.org)

[45] Zhao B, Zhong Y, Zhang L. A spectral–structural bag-of-features scene classifier for very high spatial resolution remote sensing imagery. ISPRS Journal of Photogrammetry and Remote Sensing. 2016;**116**:73-85

[46] Yu H, Yang W, Xia GS, et al. A color-texture-structure descriptor for high-resolution satellite image classification. Remote Sensing. 2016; **8**(3):259

[47] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;**521**:436-444