Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Large patch convolutional neural networks for the scene classification of high spatial resolution imagery

Yanfei Zhong
Feng Fei
Liangpei Zhang

SPIE.

Yanfei Zhong, Feng Fei, Liangpei Zhang, "Large patch convolutional neural networks for the scene classification of high spatial resolution imagery," *J. Appl. Remote Sens.* **10**(2), 025006 (2016), doi: 10.1117/1.JRS.10.025006.

Large patch convolutional neural networks for the scene classification of high spatial resolution imagery

Yanfei Zhong,[a,b,*] Feng Fei,[a,b] and Liangpei Zhang[a,b]
[a]Wuhan University, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan 430079, China
[b]Wuhan University, Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, China

Abstract. The increase of the spatial resolution of remote-sensing sensors helps to capture the abundant details related to the semantics of surface objects. However, it is difficult for the popular object-oriented classification approaches to acquire higher level semantics from the high spatial resolution remote-sensing (HSR-RS) images, which is often referred to as the "semantic gap." Instead of designing sophisticated operators, convolutional neural networks (CNNs), a typical deep learning method, can automatically discover intrinsic feature descriptors from a large number of input images to bridge the semantic gap. Due to the small data volume of the available HSR-RS scene datasets, which is far away from that of the natural scene datasets, there have been few reports of CNN approaches for HSR-RS image scene classifications. We propose a practical CNN architecture for HSR-RS scene classification, named the large patch convolutional neural network (LPCNN). The large patch sampling is used to generate hundreds of possible scene patches for the feature learning, and a global average pooling layer is used to replace the fully connected network as the classifier, which can greatly reduce the total parameters. The experiments confirm that the proposed LPCNN can learn effective local features to form an effective representation for different land-use scenes, and can achieve a performance that is comparable to the state-of-the-art on public HSR-RS scene datasets. © *2016 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JRS.10.025006]

CNN可以自动提取语义信息
而不是设计复杂的算子

但是由于数据量很小，维度小，远远小于自然场景数据，关于基于CNN的场景分类的报告较少。鉴于此，本文在现有的CNN基础上改造，提出了LPCNN

Keywords: large patch sampling; deep convolutional neural networks; remote sensing; high spatial resolution image; scene classification.

Paper 15873 received Dec. 18, 2015; accepted for publication Mar. 18, 2016; published online Apr. 25, 2016.

1 Introduction

With the increase of the spatial resolution in remote-sensing imagery, object-oriented high spatial resolution remote-sensing (HSR-RS) image classification approaches have been systematically studied and have achieved high accuracy in HSR-RS scenes.[1–4] Most object-oriented HSR-RS classification methods can quickly achieve classification map of simple and general categories that represent the physical characteristics of objects, such as vegetation and water. However, not only these physical properties are needed in HSR-RS applications, but also the higher-level semantics, such as medium residential area, golf course, and commercial streets, are increasingly required to conduct further analysis, which remains a big challenge for the object-oriented approaches. This challenge is often referred to as the "semantic gap."[5] In order to obtain higher level scene semantics, the object-oriented HSR-RS classification methods have to rely on formulated classification rules built up with laborious prior knowledge input, which cannot fit the requirements of HSR-RS images and the emergent needs of data mining in HSR-RS imagery. HSR-RS scene classification is proposed to bridge the semantic gap in HSR-RS applications.

为了获得更高层次的场景语义，面向对象的HSR-RS分类方法必须依赖于用繁琐的先验知识输入构建的公式化分类规则，这不能满足HSR-RS图像的要求和数据挖掘的紧急需求。

提出的方法

The complexity of HSR-RS scenes is increased due to the difference in illumination and the various scales of HSR-RS image scenes.[6,7] The HSR-RS scene commonly refers to a subpart of a whole remote-sensing image, which carries comparatively clear semantic information about the

*Address all correspondence to: Yanfei Zhong, zhongyanfei@whu.edu.cn

1931-3195/2016/$25.00 © 2016 SPIE

根据参考文献9，HSR-RS场景分类方法氛围两部分：
1）低级特征
2）基于语义模型方法

Earth's surface.[8] According to Ref. 9, HSR-RS scene classification approaches can be generally divided into two categories: (a) low-level features; and (b) semantic modeling-based approaches. The former approaches use global statistical information, such as the global texture or color histograms as representations of the image scenes, which are easy to compute but might be limited on large datasets.[10,11] Based on a semantic modeling strategy, the latter approaches focus on building up reliable intermediate representations, which are usually local semantic objects and concepts, to bridge the gap between the low-level features and high-level concepts.[6] Due to their excellent performances, these approaches have drawn great attention, especially the bag-of-visual-words (BoVW) model.[12–14]

BoVW分为三个步骤：
1）特征学习
2）特征表示
3）特征分类

In general, BoVW can be summarized into three major steps: (1) feature learning; (2) feature representation (using the learned feature descriptors); and (3) feature classification. The BoVW model samples an image using dense small windows that are described by various hand-crafted feature detectors, and then the image is represented as a histogram of visual word occurrences or a long feature vector via mapping the local features to a large codebook generated by sample clustering. The BoVW models have achieved many good results.[6,8,15–18]

BoVW缺陷：
因为图像被表示为局部特征的混沌集合，所以BoVW模型没有充分利用整个图像场景内的空间布局的信息。

However, because an image is represented as a chaotic collection of local features, the BoVW models do not fully exploit the information of the spatial layouts within the whole image scene.[6] Subsequently, three approaches have been proposed to mitigate the deficiency in spatial information usage. First, researchers have introduced sophisticated inference frameworks to compensate for the ambiguity raised by chaotic local features. For example, the probabilistic latent semantic analysis (pLSA) model[19,20] and the latent Dirichlet allocation (LDA) model.[21–25] These approaches have shown competitive and promising performances in HSR-RS datasets. However, the feature selection of these approaches demands a considerable amount of prior knowledge, which limits their application in other fields.[26]

pLSA 和 LDA缺陷：
需要大量的预先知识

The second kind of approaches develops empirical descriptors to utilize the spatial layout information to form improved local features. The "spatial pyramid" [or spatial pyramid matching kernel (SPMK)], which gradually partitions an image into increasingly small patches and computes histograms of local features inside patches of various scales, has shown inspiring results.[15] Reference 27 improved SPMK by calculating the spatial co-occurrence of visual words in their extended spatial co-occurrence kernel (SPCK + +) approach. However, both of these approaches need to be used with nonlinear Mercer kernels such as the intersection kernel or the chi-square kernel, where the computational complexities are high compared with linear kernels.[26] All of the aforementioned methods depend on k-means clustering to map local features into visual words, and are thus limited in their representation for classification.[28] Gabor-filter-based completed local binary patterns (GCLBP) introduces Gabor-filter-based completed binary patterns (CLBP) to strengthen the exploitation of the spatial information;[29] MS-CLBP introduces multiscale CLBP to build up the representations of the HSR-RS scene with different resolutions;[30] and the hierarchical scheme for multiple feature fusion (HMFF) employs a hierarchical multifeature fusion scheme to collect multiple feature maps as an enhanced feature representation.[31] These approaches require user expertise in the specific application and depend on too many histogram statistics, which means that they cannot be applied seamlessly in different applications.

此句说明了上段的缺陷：
1、经验
2、依赖直方图统计

The last kind of approaches relies on machine learning and optimization theory. Since both the inference framework and feature engineering of the aforementioned approaches are highly dependent on the prior knowledge or complex kernel computation, References 28 and 32 used sparse coding to improve the inference framework with scale-invariant feature transform (SIFT)-based descriptors, Reference 26 introduced the autoencoder (AE) to automatically learn local features from patches of unlabeled data detected as salient regions to replace those experimental feature descriptors; and Ref. 8 use spectral clustering to discover feature descriptors in the unsupervised feature learning framework. These methods require less prior knowledge than the previous two schemes and provide more flexibility in feature learning. As a result, they can obtain great performances based on the numerous tiny patches sampled in an HSR dataset. However, the unsupervised learning process still needs sufficient prior knowledge to design appropriate feature extractors.

无监督学习也需要足够多的先前知识

In addition, the clustering process as a key part of the unsupervised learning comes with a high computational cost. Deep learning (DL) is a set of various methods focused on learning hierarchical feature representations directly from big data with repetitive simple nonlinear neurons. DL methods have achieved state-of-the-art performances in many fields of computer vision and natural language processing.[33,34]

As one of the most popular DL models that requires less prior knowledge input, CNNs provide an end-to-end learning framework consisting of feature learning, feature coding, and classification, which are trained jointly through supervised back-propagation techniques.[35] As a result, there is no need to perform segmentation and classification on HSR-RS images to obtain simple physical categories, and to then combine the classified regions into higher scene concepts according to awkward combination rules. Fed with big data and supported by both hardware and software acceleration,[36,37] CNNs have shown their incomparable superiority in the ImageNet large-scale vision recognition challenge.[38–41] These popular CNN models are extremely deep architectures trained from millions of natural images in ImageNet; however, training CNN models for HSR remote-sensing imagery is faced with some challenges due to the unique characteristics of remote-sensing datasets. First, there are no available HSR-RS scene dataset comparable to the ones in natural scene classification. As a result, the recent published CNN architectures such as AlexNet, GoogLeNet, and VGG cannot be trained only from HSR-RS datasets, because these networks have millions of parameters that need millions of pictures to train. Furthermore, transfer learning of deep CNN models trained in natural scenes needs fine-tuning for the HSR-RS scenes, because the big datasets (with thousands of images per class) can yield an over-completed CNN model, which could probably cover most cases in HSR-RS scenes, so that the performances of the transferred CNN models should be high in most available HSR-RS scene datasets.[8,42] However, no published works have discussed the practical design of general CNN architectures that can make the best use of the currently available small HSR-RS scene datasets. The pursuit of pure high accuracy is therefore impossible for the further application of DL models in HSR-RS scene classification as well as other applications such as object detection and segmentation. Furthermore, surface objects are usually distributed randomly within an image scene, rather than the almost centrally focused natural object images, in which the center pixels have more influence on the image semantic labels, which is common in natural image scene classification. HSR-RS imagery captures a snapshot of the Earth's surface, just as cartography does, so the complex distribution patterns of surface objects in various scales are also important in HSR-RS scene classification. In this paper, we propose the large patch convolutional neural network (LPCNN) to train a CNN model from HSR-RS image dataset using improved CNN architectures to investigate the appropriate model that can balance the paradox of the deep CNN model and the limited trainable HSR-RS scene images. The major contributions of this work are as follows:

遥感数据集的特殊性？？

创新点：目前没有针对可用的小型HSR-RS场景数据集的CNN架构方法

使用改进的CNN架构从HSR-RS图像数据集训练CNN模型，以研究能够平衡深CNN模型与有限训练HSR-RS场景图像的悖论的适当模型

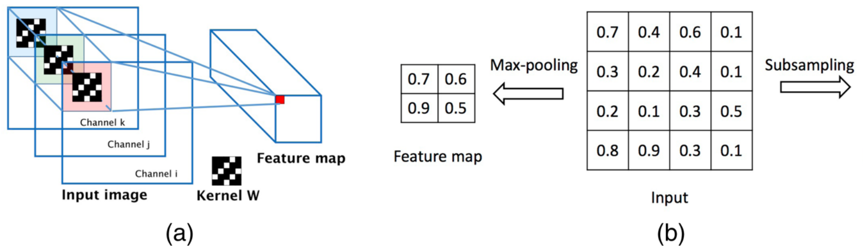
第一个创新点：采用large patch样本来弥补CNN训练集的不足
第二个创新点：改造了CNN架构，比如dropout层对HSR-RS场景分类进行了广泛的评估，用全局pooling层代替全连接层

1. Large patch sampling is proposed to generate hundreds of scene proposals for the CNN model training, in order to compensate for the shortage of the training samples in the available HSR-RS scene datasets. The influence of the sample ratio and sample amount is analyzed in detail for the different HSR-RS scene datasets, which demonstrates their importance for HSR-RS scenes of different spatial resolution.
2. An improved CNN architecture is proposed for HSR-RS scene classification. The influence of various CNN architecture designs and recent improvements, such as dropout, on HSR-RS scene classification is extensively evaluated. The global average pooling (GAP) layer is used to replace the fully connected network (FCN) as the feature classifier, which can greatly reduce the parameters in the LPCNN model and make it easier to train for the available HSR-RS scene datasets.

The rest of this paper is organized as follows. In Sec. 2, the basics of CNNs and the original CNN architecture are described. In Sec. 3, we describe the details of the LPCNN model. Section 4 provides the performance of the proposed method and the comparison with previous approaches. Section 5 discusses the influence of the hyper-parameters in the proposed method and the deep architecture design. Finally, Sec. 6 concludes the paper with a discussion of the experimental results and our future work.

2 Convolutional Neural Networks

CNNs feature stacked convolutional layers optionally followed by contrast normalization and feature pooling, corresponding to the structure of simple cells and complex cells in an animal's visual cortex.[34,43,44] The prototype of the CNN was the LeNet models featuring receptive fields



相当于average-polling

Fig. 1 (a) Convolution with weight sharing technique. All the feature maps from the last layers share the same kernel in contribution to each feature map in the next layer. (b) Feature pooling. For the same input image in the center, if max-pooling is implemented, the feature map is the one on the left; if subsampling is applied, the output feature map is the one on the right. Max-pooling keeps the maximum value within a local window, while subsampling calculates the average of the local window as its output.

and weight sharing [see Fig. 1(a)], which can greatly reduce the parameters needed to model the projection between two layers in the multilayer perceptron. Convolution neurons serve as "simple cells" while neurons in the pooling layers act as "complex cells."[43] Various pooling techniques have been discussed but max-pooling remains the most popular pooling style in modern architecture designs, as shown in Fig. 1(b).[39,41]

3 The Large Patch Convolutional Neural Network for High Spatial Resolution Imagery

In this section, we introduce the two key parts of LPCNN: stochastic large patch sampling and the small CNN architecture design. The whole learning flowchart of LPCNN is shown in Fig. 2. The necessary input image normalization is performed in the image preprocessing. The self-designed small CNN layers (convolutional layers and pooling layers) focus on feature learning and feature representation. FCN is the feature classifier in CNN, which can be replaced by GAP for parameter reduction to combat overfitting.

3.1 *Stochastic Large Patch Sampling*

BoVW and its improved approaches usually use tiny patches sampled in a dense sampling style, as shown in Fig. 3(a). This sampling strategy can precisely describe the details of objects in a scene, with the numerous patches describing their parts, corners, and intersections. However, this sampling strategy ignores the spatial relationships, for instance, the spatial distribution of different objects in a HSR-RS scene, which might be more valuable than merely object recognition because there are no strict rules about foreground/background discrimination in HSR-RS scenes. The CNN model employs a two-dimensional (2-D) convolution operation that spontaneously keeps a completed spatial relationship of each input HSR-RS patch, and it provides another

强调了空间关系的重要性

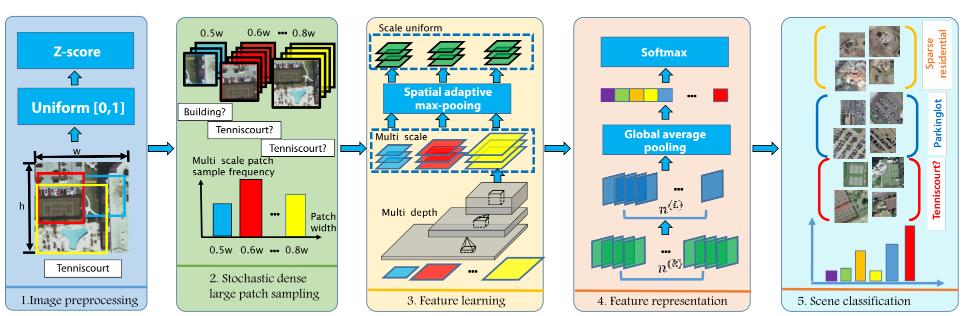
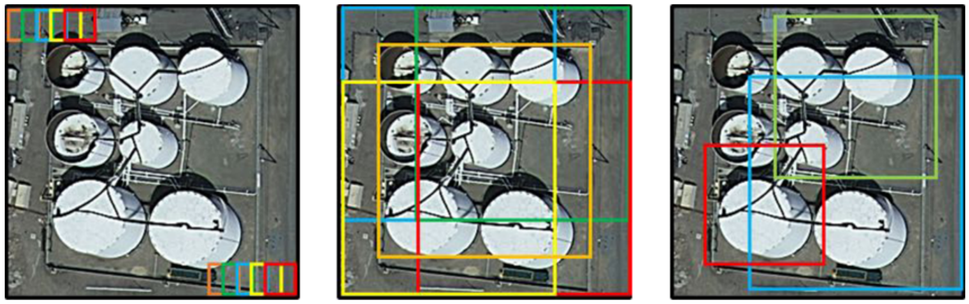


Fig. 2 Flowchart of LPCNN. The LPCNN framework consists of five parts: preprocessing, stochastic dense large patch sampling, feature learning, feature representation, and feature classification. These different parts are connected as a whole end-to-end framework.



(a) (b) (c)

Fig. 3 Different patch sampling strategies. (a) denotes the dense tiny patch sampling commonly used in BoVW and its derivative approaches; (b) shows the four corner and center large patch sampling used in AlexNet;[39] and (c) describes the stochastic large patch sampling proposed in this paper.

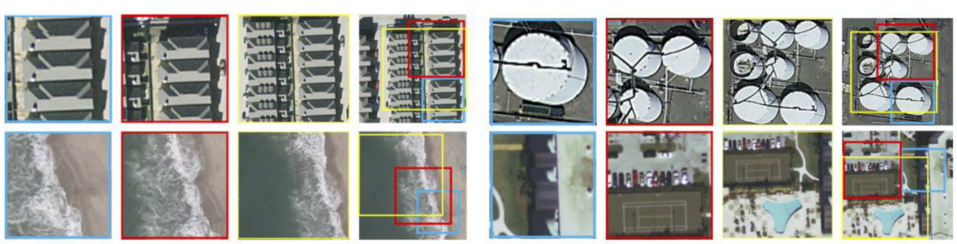
way to capture the local features of objects through deep hierarchical layers, in which the lower layers focus on learning the local features, while the higher layers generalize the processed features into global information. A more compact representation can therefore be produced to reduce the complexities left for the classifiers to handle. In other words, a CNN, as one of these deep architectures, can extract the local features and generalize the global information directly from the object level instead of the subobject parts. Figure 3(b) shows an example of the static large sampling strategy commonly used in natural image scene classification. This can effectively describe the patterns of the storage tank groups, but may dismiss the inherent structure of a single storage tank, not to mention the randomness of the object distribution. Shown as the second step in the flowchart, the LPCNN takes large patches as the input training samples to randomly capture "views" in the HSR-RS images, which are no less than a fourth of the total area of an image. The reasons for this are twofold: (a) most key objects within an HSR-RS scene are comparatively small and randomly distributed; and (b) in HSR-RS scenes, the spatial relationships help to build up more effective feature descriptors, so too small a sampling size might not be able to capture a meaningful spatial pattern.

CNN中的较低层处理了局部特征 较高层则将处理过的特征概括为全局信息

CNN作为这些深层体系结构之一，可以提取局部特征并直接从对象级别而不是子对象部分推广全局信息

HSR-RS特点：
a) HSRRS场景中的大多数关键对象相对较小且随机分布;
b) 在HSR-RS场景中，空间关系有助于构建更有效的特征描述符，因此太小的采样大小可能无法捕获有意义的空间模式

Furthermore, in order to cover the scale differences of surface objects, we propose the stochastic large patch sampling strategy denoted in the second step of the LPCNN flowchart. As shown in Fig. 3(c), for one scene image X, most of the meaningful objects in the scene can be encompassed by a large (to some extent) window according to the resolution and scale character, the ratio of which compared with the whole scene can be denoted as ψ_i in scale set $\psi = \{\psi_1, \psi_2, \ldots, \psi_k\}$, where k is the number of scale ratios. For any ψ_i, a sample window of $\psi_i h \times \psi_i w$ can be generated by randomly choosing the location of the upper left corner $\{x_{ul}, y_{ul}\}$ from $\{x, y | 0 \le x \le (1-\psi_i)h, 0 \le y \le (1-\psi_i)w, x, y \in \mathbb{N}^+, \psi_i \in \psi\}$. All these operations are denoted as $X^{(0)} = \Psi(X; m, \psi_i)$, where the output $X^{(0)}$ is the input samples for the CNN. Figure 4 shows four kinds of land-use scenes with sampling window ranging from small to large and finally those corresponding sampling window positions on the whole



(a) (b)

Fig. 4 Details of different objects in a scene from large to small scale. The upper two rows of images keep similar patterns across the different scales while the other two categories in the lower rows are very sensitive to the scale variation. (a) Object patterns that are similar across scales and (b) object patterns diverging with scale.

？？？
描述没搞懂？？？

scene, which are similar to the scale change of the sampling window in the classes of "dense residential area" and "beaches," while they are dissimilar in the "storage tanks" and "tennis court" classes.

Therefore, blindly transferring the "four corner and center" sampling strategy into HSR remote-sensing scene classification might introduce ambiguities or even confusion, and since all remote-sensing scenes out of the same dataset share identical spatial resolutions and cover a relatively limited area of the Earth's surface, a balance scale ψ_{balance} for a finite surface area can be assumed during the trade-off between in-window similarity and dissimilarity across various scales in ψ.

3.2 *Convolutional Neural Network Architecture Design for High Spatial Resolution Remote-Sensing Scene Datasets* 高分辨率遥感场景数据集的卷积神经网络体系结构设计

With the limited images in HSR-RS scene datasets, overfitting is a big challenge for CNN model training. Overfitting occurs when a model is excessively complex, and it usually results from having too many parameters relative to the number of training samples, which leads to a poor predictive performance as it can exaggerate the minor fluctuations in the data.[45] When trained from small HSR-RS scene datasets, the CNN model must control its amount of parameters in order to avoid overfitting. In HSR-RS scene classification, although the methodology about the architecture design is unclear, even in natural scene classification research, we can gather a few rules for CNN design through analyzing these published deep CNN models. The amount of parameters in CNN model is determined by two main factors: (1) the number of feature maps in the first convolutional layers and (2) the numbers of convolutional layers before the classifier layer. In order to preserve the information carried by the feature maps in the previous layers, the numbers of feature maps in the following convolutional layers are no less than the previous layer, so the number of feature maps in the first convolutional layers needs to be controlled according to the amount of images in the HSR-RS scene dataset. One of the most valuable advantages is that deep architectures can have comparable representative ability to shallow architectures with fewer parameters. As a result, the number of layers L is the other factor that balances the CNN's parameter amount and its representative ability. However, with the limited amount of training images in HSR-RS scene datasets, the balance between deep architecture and model complexity should be carefully managed.

最有价值的优势之一是深层体系结构可以具有与具有较少参数的浅层体系结构相当的代表性能力。

Not only the architecture of the deep CNN model is important in training a robust deep CNN model for HSR-RS scene classification, the basic components of the convolutional layers also have a significant influence on the performance. Recently, the rectified linear unit (ReLU) has become widely accepted as more effective nonlinear activation function for the clear form of its first derivative that is surprisingly efficient in back-propagation calculation.[39–41]

结构：
1）HSR-RS 结构要 深
2）ReLU 激励函数

3.3 *Further Tuning of the Large Patch Convolutional Neural Network Architecture with Spatial Adaptive Max-Pooling and Global Average Pooling*

用空间自适应最大池和全局平均池进一步调整大块补片卷积神经网络结构

At the end of the multiple convolutional layers, the input 2-D HSR-RS images are translated into simple but informative feature vectors in feature space. In order to obtain a more flexible dimension in classifier layers, the spatial adaptive max-pooling layer $G_{\text{SAMP}}(\cdot)$[46] is adopted for mapping the feature vectors with various dimension into the same size, which can automatically define the size of the pooling window (no larger than the input size of the feature map). For example, if the input image has a dimension of $h \times w$ and the desired feature pooling map is $h' \times w'$, the start and end positions of the pooling windows at position (i, j) in the adaptive pooling feature map are calculated according to

$$X_{\text{start}} = \left\lfloor i\frac{h}{h'} \right\rfloor, \quad X_{\text{end}} = \left\lceil (i+1)\frac{h}{h'} \right\rceil \quad Y_{\text{start}} = \left\lfloor j\frac{w}{w'} \right\rfloor, \quad Y_{\text{end}} = \left\lceil (j+1)\frac{w}{w'} \right\rceil. \tag{1}$$

In order to avoid overfitting, the dropout technique is used to perform regularization in LPCNN. Dropout has been proved to be a simple but powerful regularization technique, which randomly drops units along with their connections from the neural network during training.[45]

The GAP layer is a kind of neural network layer that can replace the FCN layer.[47] First, the output high-dimensional feature maps are used to generate one feature map for each category. Instead of flattening to a large column vector, the average of each feature map is treated as a confidence map for each category. These average values are then combined into a vector, the length of which is equal to the number of target classes. Finally, the combined confidence vector is fed directly into the SoftMax layer. The advantage is that this classifier removes the huge amount of parameters in the first two FCN layers, resulting in a dramatic parameter reduction for the CNN model.

Taking $a^{(L)}$ as the output feature representation of the convolutional layers, the GAP layer $G_{\mathrm{GAP}}(\cdot)$ is described in

$$\tilde{Y} = G_{\mathrm{GAP}}[F^{(L+1)}(a^{(L)})]. \tag{2}$$

The confidence maps generated from the GAP layers are mapped via the LogSoftMax function $\Omega(\cdot)$ to the final prediction vector $\hat{Y}$ according to Eq. (3), which is the end of the feedforward propagation. In summary, the whole prediction pipeline can be described as shown in Eqs. (4) and (5).

$$\tilde{y}_i = \Omega(\hat{y}_i) = \log \frac{e^{\hat{y}_i}}{\sum_{j=1}^{N} e^{\hat{y}_j}}, \tag{3}$$

$$\Phi(\cdot) = \underbrace{G^{(L)}(F^{(L)}(\cdots\ G^{(1)}(F^{(1)}}_{L}(\cdot))), \tag{4}$$

$$\hat{Y} = \Omega\{G_{\mathrm{GAP}}[F^{(L+1)}(G_{\mathrm{SAMP}}\{\Phi[\Psi(X;m,\psi_k);W,b]\};N)]\}. \tag{5}$$

3.4 *The Cross-Entropy Loss Function for Stochastic Gradient Descent* 随机梯度下降的交叉熵损失函数

The error between the prediction vector $\hat{Y}$ and target vector Y is used to tune the learning framework parameters via a back-propagation algorithm. Due to the available HSR-RS scene datasets being limited in the amount of training samples, stochastic gradient descent (SGD) is employed as the back-propagation method. In addition, the cross-entropy loss function is used in LPCNN training due to its fast gradient descent. Equation (6) clearly shows its computation, and its derivatives related to the model's weights W and b clearly demonstrate that the update of weights during the optimization is merely related to the differences between output $\hat{Y}$ and desired target Y. More detailed back-propagation steps can be found in Ref. 48.

$$\begin{aligned} J(Y,\hat{Y}) &= -\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)], \\ \frac{\partial J}{\partial W_j} &= -\frac{1}{N}\sum_{j}\frac{\partial \hat{y}_i}{\partial W_j}\frac{1}{\hat{y}_i(1-\hat{y}_i)}(\hat{y}_i - y_i), \\ \frac{\partial J}{\partial b_j} &= -\frac{1}{N}\sum_{j}\frac{\partial \hat{y}_i}{\partial b_j}\frac{1}{\hat{y}_i(1-\hat{y}_i)}(\hat{y}_i - y_i). \end{aligned} \tag{6}$$

4 Experiments and Analysis

We evaluated LPCNN on three different HSR remote-sensing datasets: an IKONOS land-use dataset, the UC Merced land-use dataset, and a Google Earth land-use dataset of SIRI-WHU.[14] Given that these datasets range from rather small to comparatively large datasets, we present details of the experiments and parameter analysis in order to fully evaluate the LPCNN model's performance.

4.1 *Experimental Setup*

Given that there are several parameters related to the performance of the whole LPCNN framework, we attempted to find appropriate parameters to produce the final results. In general, the ratio of training to test images was 4:1 or 80:20, and the images were randomly chosen from each category. We set a series of models with different depths, ranging from three to six, to evaluate the performance on small HSR-RS scene datasets. All the architectures of the candidate LPCNN models are listed in Table 1. The number of neurons in the input layer of the FCN classifier was 256 and the hidden layer had 128 neurons. Under the framework of Torch,[36,37] the whole framework was deployed on a SuperMicro Workstation with an NVIDIA Tesla K20 for GPU acceleration. Although there are several hyper-parameters needed for building up an LPCNN model, not all the hyper-parameters need to be tuned with different HSR-RS scene datasets. First, the hyper-parameters related to the convolutional layer are the size of convolution kernel ω and the pooling window size v, which are commonly set as 3×3 and 2×2, respectively. Second, the hyper-parameters used in SGD training are the dropout probability P_{drop} and the momentum value λ, which are frequently set as empirical values of 0.5 and 0.9, while the learning rate α should be adjusted according to the numbers of training samples. All these SGD parameters are extensively analyzed in Sec. 5. Third, the value of the patch sampling ratio ψ in the large patch sampling is the second hyper-parameter that requires additional prior knowledge of the input HSR-RS scene dataset, while the sampling frequency m is only used to neutralize the influence of the lack of sufficient training images in the HSR-RS scene dataset. Lastly, the hyper-parameter requiring prior knowledge is the number of convolutional layers L, which should be a trade-off between the amount of LPCNN model parameters and the amount of scene images from the HSR-RS images. In order to fully explore the power of the LPCNN model, we designed four kinds of LPCNN models (see Table 1) with different $L \in \{3,4,5,6\}$ to evaluate their performance when fed with various scale patches according to ψ, where $\psi \in \{0.6, 0.7, 0.8, 0.9\}$. The overall accuracy from the confusion matrix is used to assess the model performance. Details of the calculations can be found in Ref. 48.

4.2 *IKONOS Land-Use Dataset*

The IKONOS dataset is an eight-class dataset extracted from a large IKONOS scene of Wuhan acquired in 2009. The city of Wuhan is one of the biggest cities in China. The images in this dataset have four channels, with an extra near-infrared (NIR) band, which is the same kind of data as those used in other remote-sensing research. The spatial resolution is 1 m since all of the multispectral channel data have been pan-sharpened by the panchromatic channel. In this dataset, each class has 30 images in total and each image is 150×150 in dimension, which is the smallest scene dataset used in this study. Figure 5 shows typical samples of each class. Note that the images in this dataset are the same kind of HSR data as those used in other HSR-RS research, because the NIR band is included. Because this dataset has only 30 images for each category, the learning rate value was set to 0.01 and the sampling frequency was set to 100.

Table 1 The four major LPCNN architectures used for HSR scene classification.

L	Parameter layers	Architecture
3	8	64c-2p-128c-2p-256c
4	10	64c-2p-128c-2p-256c-2p-256c
5	12	64c-2p-128c-2p-256c-2p-384c-2p-256c
6	14	64c-2p-128c-2p-256c-2p-384c-2p-384c-2p-256c

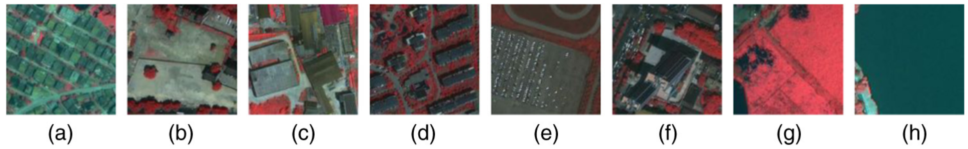


Fig. 5 Images from the IKONOS land-use dataset. Labels for the eight classes are: (a) dense building, (b) idle land, (c) industrial, (d) medium building, (e) parking-lot, (f) commercial, (g) vegetation, and (h) water.

4.2.1 *Large patch convolutional neural network performance with multiple depths and various patch sampling ratios*

Figure 6 shows the results of the LPCNN model on the IKONOS dataset. In Fig. 6(a), as the number of convolutional layers increases, the LPCNN model yields a good performance at $L = 3$ and $L = 4$, but results in model divergence because the IKONOS dataset is too small to support complex CNN model training. Figure 6(b) shows the results of the LPCNN model with different sample ratios, where $\psi = 0.7$ is the best choice for large patch sampling on the IKONOS dataset, with which the LPCNN model achieves the best performance of 92.54%. Only under this value of ψ can the LPCNN model with five convolutional layers yield an acceptable performance. The confusion matrix of the LPCNN model that yields the best performance is shown in Fig. 7, in which the two classes of "buildings" (1st class) and "vegetation" (7th class) are highly related since trees and flowers are ubiquitous in populated areas such as commercial areas, residential areas and school campuses. The second to last row indicates that the vegetation is mixed in multiple land-use scenes, and the fourth column shows that "medium building" is rather an ambiguous label due to its similarity to "parking lot," "commercial areas," and "vegetation."

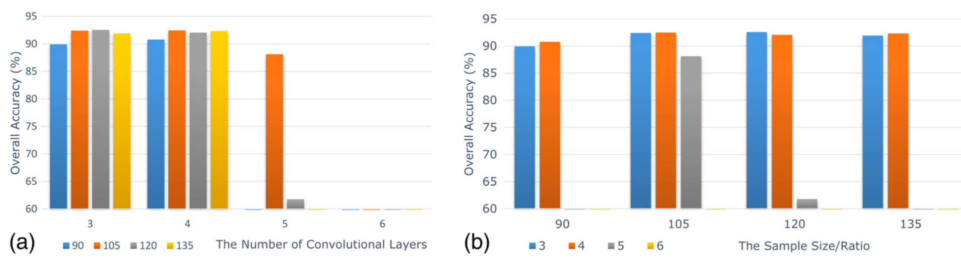


Fig. 6 Performance of the multidepth LPCNN on the IKONOS dataset: (a) LPCNN models with different numbers of convolutional layers and (b) LPCNN models with different sampling ratios ψ.

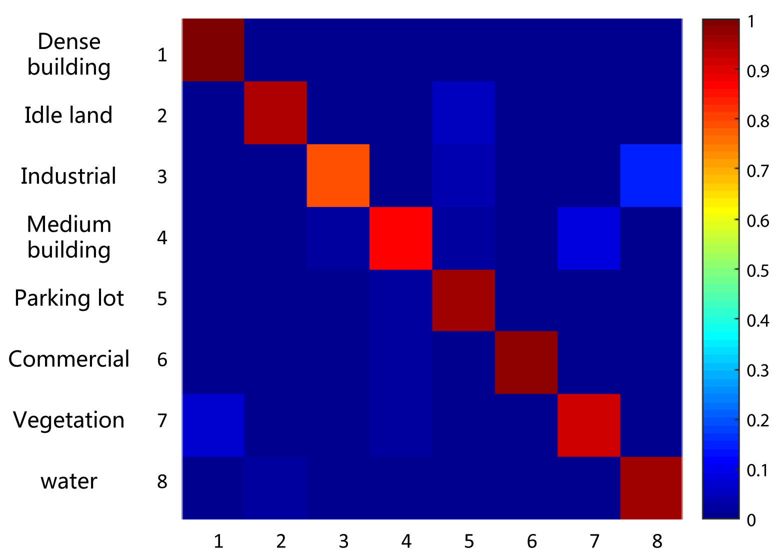


Fig. 7 The confusion matrix of the best performance of LPCNN on the IKONOS dataset.

4.2.2 *Large patch convolutional neural network compared with traditional convolutional neural network models*

Table 2 provides a comparison between proposed the LPCNN and traditional CNN models in architecture design and classification performance under two different value of the learning rate α. When $\alpha = 0.001$, the LPCNN model takes small steps during the optimization, allowing a finer performance comparison and trend analysis. TF-CNN, which achieves an accuracy of 75.94%, is made up of the tanh activation function and FCN, and is a prototype of the traditional CNN. Since the ReLU performed better in back propagation in Ref. 39, RF-CNNs were built and evaluated for three different combinations of convolution and pooling configuration, achieving OAs of 63.44%, 72.93%, and 76.98%, respectively. The convolutional kernel size was reduced from five to three in order to explore the local details/features in smaller areas, and dimension reduction in the pooling layers leads to better generalization, and it large window feature pooling in higher layer is better than in lower layers due to more distinguishable and robust features are formed in higher layers. The dropout technique is an effective way to regularize models,[45] which improves the accuracy by about 0.5% so that RDF-CNN achieves an accuracy of 77.46%. Since the SAMP layer is free from the dimension of the input feature maps, RDSF-CNN is more flexible in lower-layer feature learning, and achieves an accuracy of 80.21%. Since they are very close in network architectures, RDF-CNN and RDSF-CNN are bound to obtain comparable results. Due to the fact that GAP is a kind of classifier with no parameters to optimize, RDSG-CNN achieves the best performance of 88.29%, and GAP is also used in the LPCNN architecture for its advantages and performance in large scene patch feature learning and classification. When $\alpha = 0.01$, the LPCNN model takes an aggressive large step when finding the global minima. Compared with a small value of α, all the CNN models yield a better performance with $\alpha = 0.01$. However, the large α leads to severe model divergence in the architecture consisting of dropout and the FCN classifier, shortly after the best performance, which probably suggests that the number of the neurons in the FCN hidden layer (currently 128) has reached the tipping point for CNN architectures without dropout. Therefore, we doubled the number of the neurons in the FCN hidden layer to perform the experiment, in which severe model divergence quickly happens after achieving the presented performance. In this case, the α and dropout in the FCN-based CNN should be more carefully compared. With the FCN classifier replaced by GAP, the LPCNN architecture is robust and effective in both empirical learning rates, and it yields the best performance on the IKONOS dataset, an OA of 92.54%.

4.3 *UC Merced Land-Use Dataset*

The UC Merced land-use dataset, which is a 21-class remote-sensing data collected from large optical images (with RGB color space) of the US Geological Survey, covers various regions of the United States. Its original images are RGB color images with a one foot (~0.3 m) spatial

Table 2 Performance of different CNN models on the IKONOS dataset [OA (%)].

Model	Key parts	Architecture	$\alpha = 0.001$	$\alpha = 0.01$
TF-CNN	Tanh/FCN	5c-2p-5c-2p-3c-3p	75.94	76.63
RF1-CNN	ReLU/FCN	5c-2p-5c-2p-3c-3p	63.44	81.94
RF2-CNN	ReLU/FCN	3c-5p-3c-2p-3c-3p	72.93	81.48
RF3-CNN	ReLU/FCN	3c-3p-3c-2p-3c-6p	76.98	83.27[a]
RDF-CNN	ReLU/Dropout/FCN	3c-2p-3c-2p-3c-6p	77.46	84.31[b]
RDSF-CNN	ReLU/Dropout/SAMP/FCN	3c-2p-3c-2p-3c-/	80.21	81.79
RDSG-CNN	ReLU/Dropout/SAMP/GAP	3c-2p-3c-2p-3c-/	88.29	92.54

[a]The FCN classifier is 256-256-8.
[b]The FCN classifier is same as the RDF-CNN.

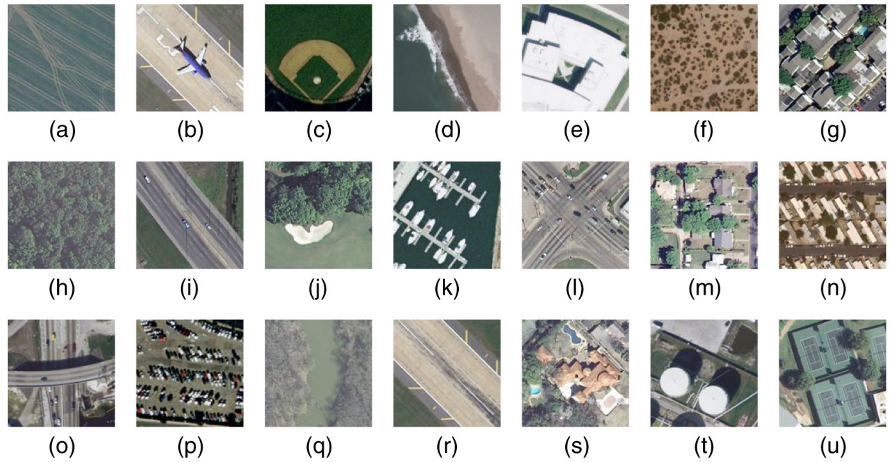


Fig. 8 Images from the UC Merced land-use dataset. The 21 class labels are: (a) agriculture, (b) airplane, (c) baseball diamond, (d) beach, (e) buildings, (f) chaparral, (g) dense residential, (h) forest, (i) freeway, (j) golf course, (k) harbor, (l) intersection, (m) medium residential, (n) mobile home park, (o) overpass, (p) parking lot, (q) river, (r) runway, (s) sparse residential, (t) storage tanks, and (u) tennis court.

resolution. Figure 8 shows images randomly chosen from each class. In each class, there are 100 images with a dimension of 256×256. More information is available in Ref. 27. This dataset is the highest spatial resolution data used in this study, in which the whole region of most scenes is roughly occupied by surface objects similar to natural scenes, but there are still a few discrete distributed cases. In this dataset, the learning rate value was set to 0.001 for a slightly slower but finer parameter training and the sample frequency was set to 100.

这部分在教如何调参

4.3.1 *Large patch convolutional neural network performance with multiple depths and various patch sampling ratios*

Figure 9 shows the results of the LPCNN model on the UC Merced land-use dataset. As the number of convolutional layers increases in Fig. 9(a), the performances of the LPCNN model decreases, which suggests that the UC Merced dataset can support complex CNN model training, but it is still limited in data capacity for deeper CNN model training. Figure 9(b) shows the performance of the LPCNN model with different sample ratios ψ. The samples generated under $\psi = 0.7$ can support more complex LPCNN models while the best performance is achieved by the three convolutional layer LPCNN with the largest ψ value (an OA of 89.90%), which suggests that this LPCNN architecture is more appropriate for the current UC Merced dataset, but the LPCNNs with more convolutional layers are also promising if more training images are added.

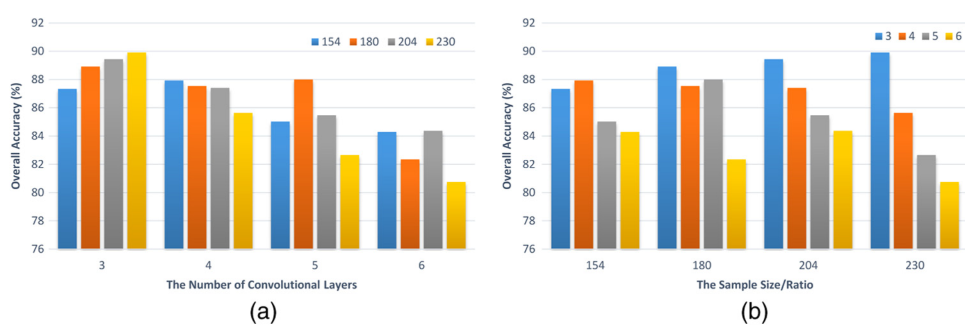


Fig. 9 Performance of the multidepth LPCNN on the UC Merced dataset: (a) LPCNN models with different numbers of convolutional layers and (b) LPCNN models with different sampling ratios ψ.

Table 3 LPCNN/RDSG-CNN performance compared with traditional CNN models on the UC Merced dataset.

Model	Key parts	Architecture	OA(%)
TF-CNN	Tanh/FCN	5c-2p-5c-2p-5c-3p	78.98
RF1-CNN	ReLU/FCN	5c-2p-5c-2p-5c-3p	80.63
RF2-CNN	ReLU/FCN	3c-6p-3c-3p-3c-4p	82.25
RF3-CNN	ReLU/FCN	3c-3p-3c-2p-3c-15p	83.78
RDF-CNN	ReLU/Dropout/FCN	3c-2p-3c-2p-3c-15p	85.79
RDSF-CNN	ReLU/Dropout/SAMP/FCN	3c-2p-3c-2c-3p-/	85.66
RDSG-CNN	ReLU/Dropout/SAMP/GAP	3c-2p-3c-2p-3c-/	89.90

4.3.2 *Large patch convolutional neural network compared with traditional CNN models*

As with the IKONOS dataset, a comparison between LPCNN and the traditional CNN is presented in Table 3, where the results show a similar trend to the IKONOS dataset, which suggests that CNNs/LPCNNs can learn robust scene features in similar framework. The original TF-CNN obtains an OA of 78.98%, which serves as a baseline for the following CNN models. The RF-CNNs achieve OAs of 80.63%, 82.25%, and 83.78% with the different network architecture designs. RDF-CNN and RDSF-CNN yield OAs of 85.79% and 85.66%. RDSG-CNN/LPCNN achieves the best performance: 89.90%.

4.3.3 *Large patch convolutional neural network compared with published works*

Since the UC Merced is an open-access land-use dataset that serves as a benchmark in HSR scene classification, Table 4 shows a comparison with the other published approaches. The first three experiments involved topic model approaches, i.e., BoVW, pLSA, and LDA. Using statistical features such as the mean and standard deviation of the spectra, the topic models obtain OAs of 72.05%, 80.71%, and 81.92%, respectively. Reference 15 introduces the SPMK to the spatial layout pattern in the scene to extend the basic features of BoVW; the SPMK is inspiring in spatial layout information utilization, but it still belongs to the low-level feature descriptors, which achieved an accuracy of 74.00%. The method proposed in Ref. 27 extends SPMK via the introduction of co-occurrence calculation of the spatial features, obtaining an accuracy of 76.05%. Distinct from feature engineering, Reference 49 combines SIFT descriptors with sparse coding to build up and over-complete dictionary for the low-level features, which achieves an accuracy of 81.67%. Reference 26 introduces an AE trained with sparse salient patches to replace the dense SIFT descriptors used in Ref. 49 to achieve features totally formulated by the dataset itself, achieving an accuracy of 82.72%. Reference 25 proposed semantic allocation level (SAL) multifeature fusion strategy based on probabilistic topic model (PTM), which

Table 4 Comparison in performance with the UC Merced dataset and other published approaches.

Methods	BoVW	pLSA	LDA	SPM[15]	SPCK++[27]	SIFT+SC[49]	S-UFL[26]
OA (%)	72.05	80.71	81.92	74.00	76.05	81.27	82.72
Methods	SAL-PTM[25]	TF-CNN	LPCNN	GCLBP[29]	MS-CLBP1[30]	MS-CLBP2[30]	HMFF[31]
OA (%)	88.33	78.98	89.90	90.00	90.60	89.90	92.38

introduces topic-level clustering to model the relationships of the scenes, yielding an accuracy of 88.33%. The LPCNN model yielded an OA of 89.90%. Based on sophisticated LBP features, the GCLBP approach achieves an OA of 90.0%; The two different implementations of the MS-CLBP methods achieve OAs of 90.60% and 89.90%, respectively. HMFF achieves an OA of 92.38%. LPCNN uses a CNN models that is specially designed for small HSR-RS scene datasets, and a large patch sampling technique to generate hundreds of scene proposals for CNN training, achieving an OA of 89.90%. The HMFF method employs multiple features from different kinds of hand-crafted feature descriptors so that this approach is rather similar to the boosting methods based on multiple models, while the previous methods, including LPCNN, report the performance of a single model. Although it does not achieve the best accuracy among these methods, the LPCNN model still obtains a comparable performance to these carefully designed low-level feature approaches, which demonstrates that the proposed LPCNN model is effective in HSR-RS scene classification.

Furthermore, Fig. 10 shows the confusion matrix of the best LPCNN model. Since it is complicated to check directly from this visualized matrix, we presented another comparison with the state-of-the-art approaches in Fig. 11, in which the blue color represents the GCLBP methods, the two green colors represent two different implementation of the MS-CLBP approaches, and

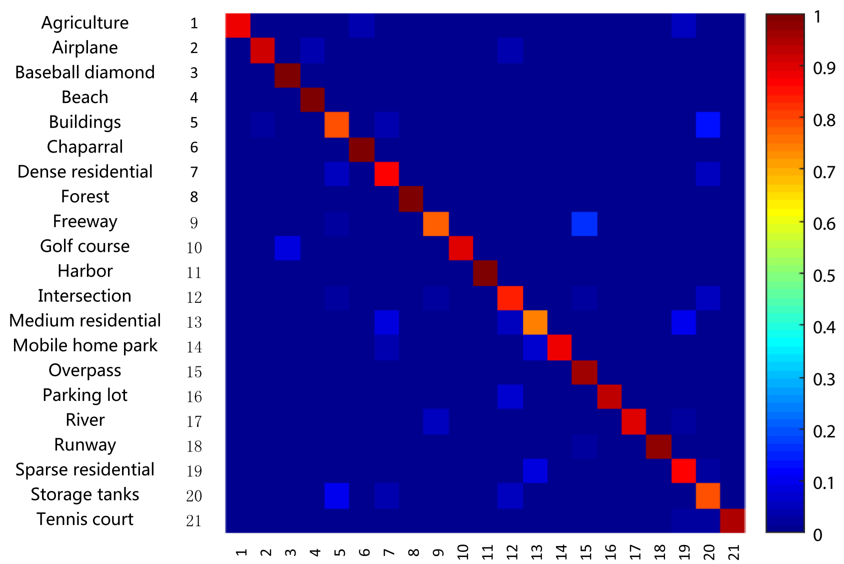


Fig. 10 Confusion matrix of best LPCNN model on the UC Merced dataset.

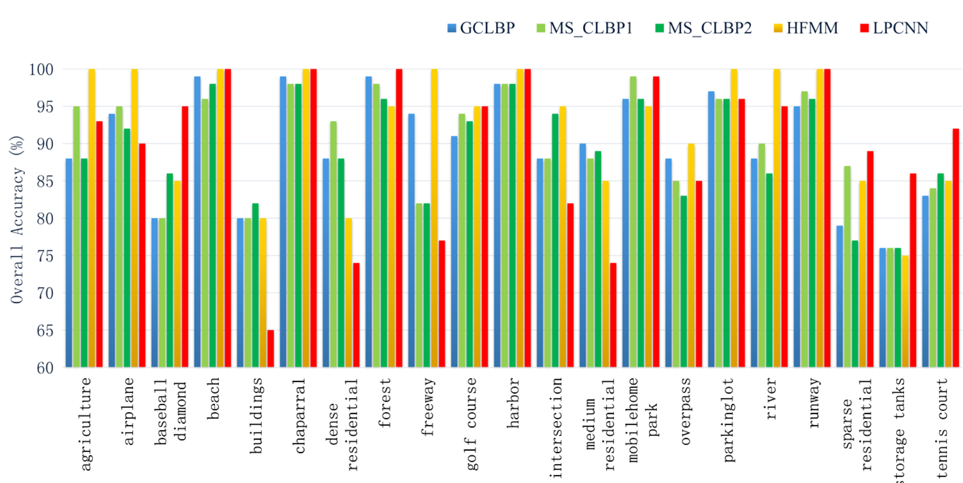


Fig. 11 The performance of the LPCNN model compared with the state-of-the-art approaches by each category.

the red color represents the LPCNN model. The LPCNN model achieves a lower OA than these methods because it performs very poorly in telling the difference between several groups: (a) “buildings,” “dense residential,” and “medium residential;” and (b) “freeway,” “intersection,” and “overpass.” The HSR-RS scenes belonging to each group have many identical features, and determining their labels depends not only on the spatial features in the image, but also on the influence of the location, city development, and government policy, so it is also a challenge for human beings to explicitly tell the difference. However, the poor performance on these categories reveals that LPCNN does extract the intrinsic features from the input HSR-RS images, and gradually generalized them into higher level information. High-level information is robust and demanded in HSR-RS application, but low-level features are important and effective as well, at least in current small HSR-RS scene datasets. Given low-level features are combined with higher level information, the LPCNN model would probably achieve a more convincing performance. Except for these categories, the LPCNN model achieves the best performance in categories such as “beach,” “forest,” “river,” and “harbor.” Especially in the “storage tanks” and “tennis court” classes, the LPCNN model presents a dominant performance, and even the fusion of multiple hand-crafted features cannot yield a comparable result, which again demonstrates the superiority of the LPCNN model in HSR-RS scene datasets.

4.4 *Google Earth Land-Use Dataset of SIRI-WHU*

The Google Earth land-use dataset designed by Intelligent Data Extraction and analysis of Remote Sensing (RSIDEA) Group in Wuhan University (SIRI-WHU),[14] with 12 different kinds of scenes gathered from Google Earth images covering urban areas of China, has a 2-m spatial resolution, as shown in Fig. 12. Focused on the Chinese landscape, this dataset has several unique categories, such as “dense residential” and “medium residential,” and “medium residential” might be classified as “dense residential” in other land-use datasets. For each class, there are 200 images with the size of 200×200, which offers more samples than both the previous two datasets. In this dataset, the learning rate value α was set in the same way as the experiments on the UC Merced land-use dataset.

4.4.1 *Large patch convolutional neural network performance with multiple depths and various patch sampling ratios*

The results of the LPCNN model on the Google Earth dataset are shown in Fig. 13. Figure 13(a) shows that the performances gradually decreased with the increase of L, which also suggests that the Google Earth dataset of SIRI-WHU is insufficient in data amount to train an over-complete CNN model for HSR-RS scene classification. Interestingly, when $\psi = 0.6$, the architecture of $L = 4$ outperforms that of $L = 3$, which demonstrates that the CNN models can hierarchically generalize features to form robust feature representations. Figure 13(b) shows the performances

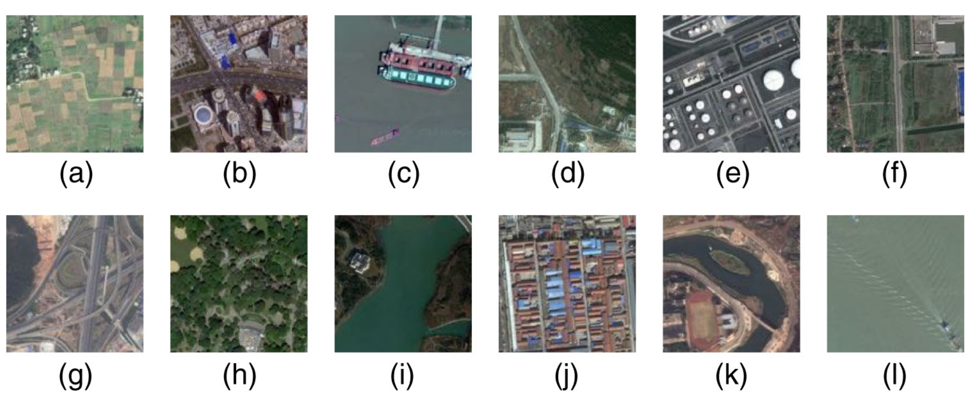


Fig. 12 Images from Google Earth Land-use Dataset of SIRI-WHU. The labels of 12 categories are: (a) agriculture, (b) commercial, (c) harbor, (d) idle land, (e) industrial, (f) meadow, (g) overpass, (h) park, (i) pond, (j) residential, (k) river, and (l) waterbody.

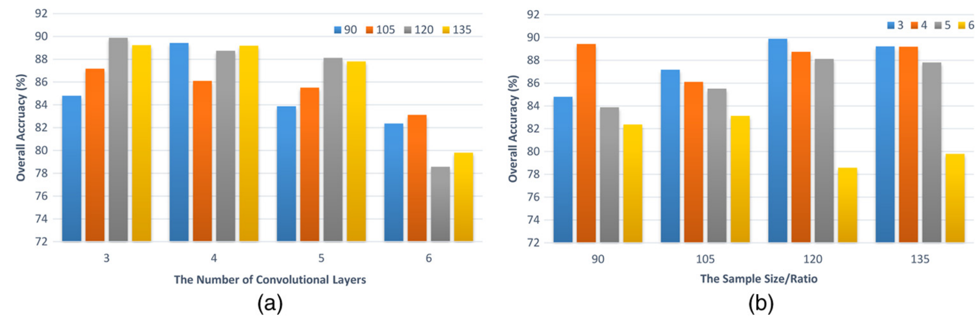


Fig. 13 Performance of the multidepth LPCNN on the Google Earth dataset of SIRI-WHU: (a) LPCNN models with different numbers of the convolutional layers and (b) LPCNN models with different sampling ratios ψ.

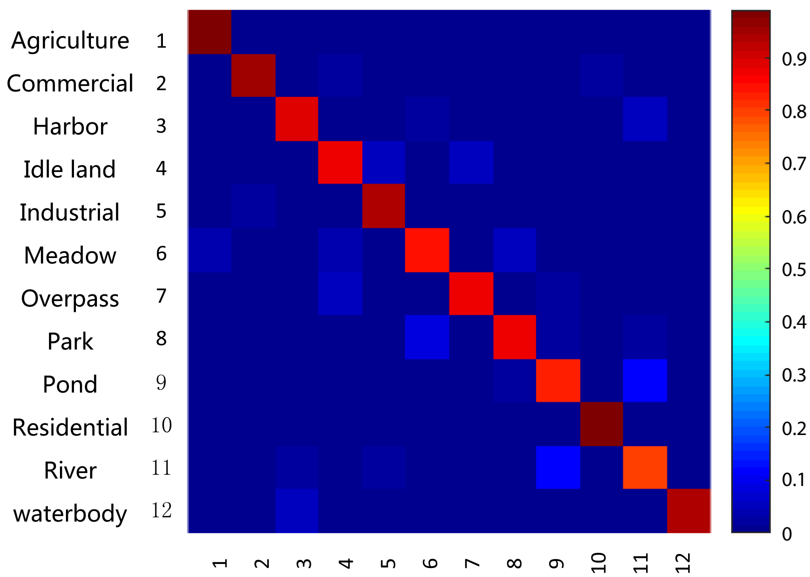


Fig. 14 Confusion matrix of best LPCNN model on the Google Earth dataset of SIRI-WHU.

across different sample ratios ψ, which suggests that $\psi = 0.8$ and $\psi = 0.9$ are better choices when training the LPCNN models. The architecture determined by $L = 3$ and $\psi = 0.8$ achieves the best performance, an OA of 89.88%. Figure 14 shows confusion matrix of best LPCNN model, where it can be seen that river class (11th class) and pond class (9th class) are too similar for the LPCNN model to classify.

4.4.2 *Large patch convolutional neural network compared with the traditional CNN models*

The performances of the traditional CNN and the LPCNN model are presented in Table 5, where TF-CNN obtains an accuracy of 82.81%. The RF-CNNs achieve gradually increasing performances of 79.83%, 78.73%, and 81.03%, with smaller convolutional kernels and high-layer larger pooling windows. The dropout technique improves the performance to 82.56%, and RDF-CNN using the SAMP layer generates a slightly higher accuracy of 84.10%. Finally, using GAP to replace FCN, RDSG-CNN/LPCNN obtains an accuracy of 89.88%, which is the best performance among all the CNN models.

5 Parameter Analysis

In this section, we analyze the hyper-parameters, which were set as constants when comparing LPCNN with different values of network depth L and sampling ratio k. The IKONOS dataset

Table 5 LPCNN/RDSG-CNN performance compared with traditional CNN on the Google dataset of SIRI-WHU.

Model	Key parts	Architecture	OA(%)
TF-CNN	Tanh + FCN	5c-2p-5c-2p-3c-3p	82.81
RF1-CNN	ReLU + FCN	5c-2p-5c-2p-3c-3p	79.83
RF2-CNN	ReLU + FCN	3c-5p-3c-2p-3c-4p	78.73
RF3-CNN	ReLU + FCN	3c-3p-3c-2p-3c-10p	81.03
RDF-CNN	ReLU + Dropout + FCN	3c-2p-3c-2p-3c-10p	82.56
RDSF-CNN	ReLU + Dropout + SAMP + FCN	3c-2p-3c-2p-3c-/	84.10
RDSG-CNN	ReLU + Dropout + SAMP + GAP	3c-2p-3c-2p-3c-/	89.88

was used to evaluate these hyper-parameters since it is the same kind of HSR-RS scene data as those used in other HSR-RS research. According to the previous performance matrix, the LPCNN model with $L = 4$ at scale $\psi = 0.7$ was used as the reference model to analyze the sample frequency m, the dropout probability P_{drop}, the learning rate α, and the momentum value λ, which was also used to demonstrate the effectiveness of the empirical values of several hyper-parameters that are free from fine-tuning with specific HSR-RS scene datasets.

5.1 *Sampling Frequency of the Stochastic Large Patch Sampling*

In the previous model training, the total amount of samples m of all the iterations was kept the same as 20 epochs of 100 samples, i.e., 2000 input images in total, unless overfitting of the signal was asserted. For 10, 20, 30, 40, 50, 80, 100, 150, and 200 samples for each input images, the maximum training epoch was set to 15. As shown in Fig. 15(a), when the sampling amount is 100 and 150, LPCNN obtains the best performance. Given the computational efficiency, 100 is more preferable since 150 requires more sampling operations with few improvements in performance.

选择100次迭代更合适
因为尽管150次的有所改善
但是所需要的样本也会增加
更多

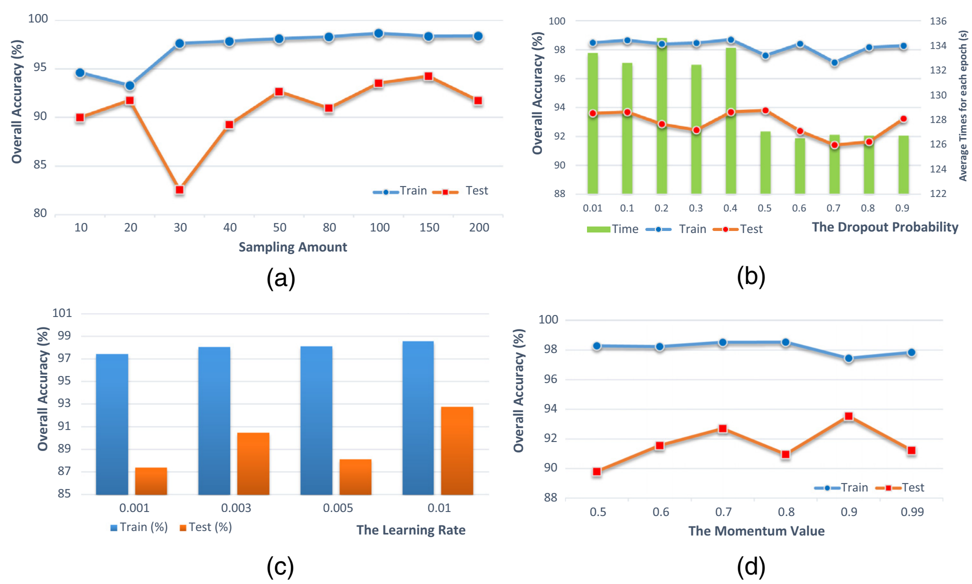


Fig. 15 (a) The best performance of LPCNN under different amount of stochastic dense large patch sampling. (b) The best performance of LPCNN under different dropout values. (c) The best performance of LPCNN under various learning rates. (d) Best performance of LPCNN under different momentum values.

5.2 *Hyper-Parameters of the Network Architecture*

In LPCNN, most of the hyper-parameters, such as the learning rate, momentum, and probability value of dropout, are kept as empirical values.

5.2.1 *The dropout probability*

The dropout technique is employed to regularize the whole LPCNN model. The dropout probability P_{drop} is often set as 0.5, but other dropout probabilities, such as 0.3, 0.4, 0.6, and 0.7, are extensively evaluated here. In Fig. 15(b), the empirical dropout value of 0.5 yields a comparable performance and reaches a balance between accuracy and time efficiency. When P_{drop} is a small value (0.01 and 0.1), LPCNN can achieve a comparable performance but needs more time during each training epoch. When the dropout probability is rather high (from 0.7 to 0.9), the results fluctuate because most of the features are dropped, so that LPCNN cannot gather useful information to build up a robust feature representation.

合理设置dropout

5.2.2 *Learning rate for stochastic gradient descent*

As for SGD, a proper learning rate α accelerates the training, and it is often recommended to select from [0.1, 0.001] in the experiments.[50] In Fig. 15(c), the learning rate of 0.001 obtains a lower accuracy than 0.01, because the searching step in the optimization is much smaller, and if more iterations are allowed, both learning rates could reach comparable performances. Consequently, they are all appropriate for LPCNN training but the former is a bit slow on the IKONOS dataset. A value of larger than 0.01 is not presented in the figure because the LPCNN model diverges under these values of learning rate. As a result, the empirical values of the learning rate, i.e., 0.01 and 0.001 are both effective for HSR-RS scene classification.

调节学习率参数

5.2.3 *Momentum for stochastic gradient descent*

In order to prevent the model from getting stuck in local minima and to approach the global minima as soon as possible, a momentum technique is used, which is often set as 0.7, 0.8, 0.9, or 0.99. An evaluation of these momentum values is shown in Fig. 15(d). When $\lambda = 0.9$, LPCNN achieves its best performance, whereas the results of the other values are more unstable, because using an aggressive small λ can might misguide the search direction during the optimization, and a large λ value makes the parameters too slow to approach the global minimal point. As a result, an empirical momentum value of 0.9 is more reliable.

6 Conclusion

In this paper, we have proposed an effective CNN model named the large patch convolution neural network (LPCNN) model for the scene classification of the available small HSR-RS scene datasets. In order to overcome the challenges caused by the small dataset capacity and the inherent complexity of HSR-RS scenes, a stochastic large patch sampling strategy is introduced to fully exploit the spatial relationships in HSR-RS scenes. Second, we employ SAMP and GAP layers to build up a simplified CNN architecture that is more appropriate for the available small HSR-RS scene datasets, including improved convolutional layer components as the ReLU activation function, and a smaller sliding window size. The proposed method is able to yield reliable OA for remote-sensing scene datasets. Experiments were performed on three different HSR-RS scene datasets with different amount of scene patches and different spatial resolutions. In terms of learning merely from training set images and a single model, LPCNN is able to yield comparable results to the state-of-the-art methods, and can reveal the HSR-RS scene dataset's bottleneck caused by the information discrepancy between the training set and test set. Since the model performance is not only determined by the approach itself, but is also significantly influenced by training images from the HSR-RS scene datasets, it may be that LPCNNs are not superior to hand-crafted feature descriptors on small HSR-RS scene datasets, but identical

本文所做的工作：
1）提出了随机大样本块策略
2）采用SAMP和GAP层重构了CNN，包括采用ReLU激励函数和小滑动窗口

LPCNN architectures can be applied to different HSR-RS scene datasets, with almost no structural change, and can yield an OA that is comparable to the previous methods based on well-designed hand-crafted feature descriptors. Consequently, in our future work, in order to build up a more reliable CNN model, we will re-evaluate the value of the low-level features and attempt to discover an efficient way to resolve the complex categories where the LPCNN does not perform well. We will also try to explore adaptive large patch sampling ratio method to build up a more flexible training patch sampling methods for a more heuristic exploitation of the available small HSR-RS scene datasets.

Acknowledgments

The authors would like to thank the editor, associate editor, and anonymous reviewers for their helpful comments and advice. The authors would also like to thank the developers in the Torch developer community for their guidance and help on GPU acceleration. This work was supported by National Natural Science Foundation of China under Grant No. 41371344, State Key Laboratory of Earth Surface Processes and Resource Ecology under Grant No. 2015-KF-02, Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT1278, Natural Science Foundation of Hubei Province under Grant No. 2015CFA002, and Open Research Fund Program of Shenzhen Key Laboratory of Spatial Smart Sensing and Services (Shenzhen University).

References

1. T. Blaschke, "Object based image analysis for remote sensing," *ISPRS J. Photogramm. Remote Sens.* **65**(1), 2–16 (2010).
2. X. Zhang, B. Wu, and Q. Li, "Research on classification of high-resolution remote sensing image," *Proc. SPIE* **6786**, 67862Z (2007).
3. X. Wang, X. Chen, and M. Li, "Classification of high spatial resolution remote sensing image using SVM and local spatial statistics Getis-Ord Gi," *Proc. SPIE* **8006**, 80060N (2011).
4. Z. Chen et al., "High resolution remote sensing image classification with multiple classifiers based on mixed combining rule," *Proc. SPIE* **6788**, 67881J (2007).
5. A. Hauptmann et al., "Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news," *IEEE Trans. Multimedia* **9**, 958–966 (2007).
6. L. Zhou, Z. Zhou, and D. Hu, "Scene classification using a multi-resolution bag-of-features model," *Pattern Recognit.* **46**(1), 424–433 (2013).
7. J. Qin and N. H. Yung, "Scene categorization via contextual visual words," *Pattern Recognit.* **43**(5), 1874–1888 (2010).
8. F. Hu et al., "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **8**, 2015–2030 (2015).
9. A. Bosch, X. Muñoz, and R. Martí, "Which is the best way to organize/classify images by content?," *Image Vision Comput.* **25**(6), 778–791 (2007).
10. A. Vailaya, A. Jain, and H. J. Zhang, "On image classification: city vs. landscape," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 3–8, IEEE (1998).
11. N. Serrano, A. E. Savakis, and J. Luo, "Improved scene classification using efficient low-level features and semantic cues," *Pattern Recognit.* **37**(9), 1773–1784 (2004).
12. B. Zhao et al., "The Fisher Kernel coding framework for high spatial resolution scene classification," *Remote Sens.* **8**(2), 157 (2016).
13. B. Zhao, Y. Zhong, and L. Zhang, "A spectral–structural bag-of-features scene classifier for very high spatial resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.* **116**, 73–85 (2016).
14. B. Zhao et al., "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.* **54**(4), 2108–2123 (2016).

15. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 2169–2178, IEEE (2006).
16. J. Yang et al., "Evaluating bag-of-visual-words representations in scene classification," in *Proc. of the Int. Workshop on Multimedia Information Retrieval*, pp. 197–206, ACM (2007).
17. D. Gökalp and S. Aksoy, "Scene classification using bag-of-regions representations," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE (2007).
18. Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. of the 18th SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (GIS '10)*, pp. 270–279, ACM (2010).
19. A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *Computer Vision–ECCV 2006*, pp. 517–530, Springer (2006).
20. G. Cheng et al., "Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA," *Int. J. Remote Sens.* **34**(1), 45–59 (2013).
21. D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.* **3**, 993–1022 (2003).
22. L. Cao and L. Fei-Fei, "Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes," in *IEEE 11th Int. Conf. on Computer Vision*, pp. 1–8, IEEE (2007).
23. M. Liénou, H. Maître, and M. Datcu, "Semantic annotation of satellite images using latent Dirichlet allocation," *IEEE Geosci. Remote Sens. Lett.* **7**(1), 28–32 (2010).
24. C. Vaduva, I. Gavat, and M. Datcu, "Latent Dirichlet allocation for spatial analysis of satellite images," *IEEE Trans. Geosci. Remote Sens.* **51**(5), 2770–2786 (2013).
25. Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.* **53**, 6207–6222 (2015).
26. F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.* **53**(4), 2175–2184 (2015).
27. Y. Yang and S. Newsam, "Spatial pyramid co-occurrence for image classification," in *IEEE Int. Conf. on Computer Vision (ICCV '11)*, pp. 1465–1472, IEEE, Barcelona (2011).
28. A. Cheriyadat, "Aerial scene recognition using efficient sparse representation," in *Proc. of the Eighth Indian Conf. on Computer Vision, Graphics and Image Processing*, p. 11, ACM (2012).
29. C. Chen et al., "Gabor-filtering-based completed local binary patterns for land-use scene classification," in *IEEE Int. Conf. on Multimedia Big Data (BigMM)*, pp. 324–329 (2015).
30. C. Chen et al., "Land-use scene classification using multi-scale completed local binary patterns," *Signal Image Video Process.* **10**(4), 745–752 (2015)
31. W. Shao et al., "A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization," in *Computer Vision Systems*, pp. 324–333, Springer, Switzerland (2013).
32. G. Sheng et al., "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *Int. J. Remote Sens.* **33**(8), 2395–2412 (2012).
33. Y. Bengio, "Learning deep architectures for AI," *FNT Mach. Learn.* **2**(1), 1–127 (2009).
34. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
35. L. Zhang et al., "Deep learning for remote sensing image understanding," *J. Sensors* **501**, 173691 (2015).
36. R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: a MATLAB-like environment for machine learning," in *BigLearn, NIPS Workshop, (EPFL-CONF-192376)* (2011).
37. S. Chetlur et al., "CuDNN: efficient primitives for deep learning," *CoRR*, arXiv preprint arXiv:1410.0759 (2014).
38. J. Deng et al., "Imagenet: a large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE (2009).
39. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012).

40. C. Szegedy et al., "Going deeper with convolutions," arXiv preprint arXiv:1409.4842 (2014).
41. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, arXiv preprint arXiv:1409.1556 (2014).
42. M. Castelluccio et al., "Land use classification in remote sensing images by convolutional neural networks," *CoRR*, arXiv preprint arXiv:1508.00092 (2015).
43. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.* **160**(1), 106–154 (1962).
44. K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.* **36**(4), 193–202 (1980).
45. N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).
46. K. He et al., "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision–ECCV 2014*, pp. 346–361, Springer (2014).
47. M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, abs/1312.4400 (2013).
48. Y. A. LeCun et al., "Efficient backprop," in *Neural Networks: Tricks of the Trade*, pp. 9–48, Springer (2012).
49. A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.* **52**(1), 439–451 (2014).
50. T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 343–351, arXiv preprint arXiv:1206.1106 (2012).

Yanfei Zhong received a BS degree in information engineering and a PhD degree in photogrammetry and remote sensing from Wuhan University, China, in 2002 and 2007, respectively. He has been with the LIESMARS, Wuhan University, since 2007 and is currently a full professor. His research interests include multi- and hyperspectral remote sensing data processing, high resolution image processing and scene analysis, and computational intelligence.

Feng Fei received a BS degree in geographic information science from Nanjing Normal University, Nanjing, China, in 2013. He is currently working toward an MS degree in photogrammetry and remote sensing in the State Key Laboratory of Information Engineering in LIESMARS, Wuhan University, Wuhan, China. His major research interests include scene understanding for high spatial resolution remote sensing imagery and deep learning.

Liangpei Zhang received a BS degree in physics from Hunan Normal University, Changsha, China, in 1982, an MS degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and a PhD degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998. He is currently the head of the remote sensing division, LIESMARS, Wuhan University. He is also a "Chang-Jiang Scholar" chair professor.