Letter

A Fast Deep Perception Network for Remote Sensing Scene Classification

Ruchan Dong [1,2,3,*], Dazhuan Xu [1], Lichen Jiao [4], Jin Zhao [4] and Jungang An [5]

1 College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; xudazhuan@nuaa.edu.cn
2 School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China
3 Software Testing Engineering Laboratory of Jiangsu Province, Nanjing 211169, China
4 Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China; lchjiao@mail.xidian.edu.cn (L.J.); zhaojin119@stu.xidian.edu.cn (J.Z.)
5 Information Technology Section of Shanghai Railway Administration, Jinhua 322001, China; jgan@haimo.com.cn
* Correspondence: ruchandong@nuaa.edu.cn; Tel.: +86-137-7057-0313

Received: 24 January 2020; Accepted: 20 February 2020; Published: 22 February 2020

Abstract: Current scene classification for high-resolution remote sensing images usually uses deep convolutional neural networks (DCNN) to extract extensive features and adopts support vector machine (SVM) as classifier. DCNN can well exploit deep features but ignore valuable shallow features like texture and directional information; and SVM can hardly train a large amount of samples in an efficient way. This paper proposes a fast deep perception network (FDPResnet) that integrates DCNN and Broad Learning System (BLS), a novel effective learning system, to extract both deep and shallow features and encapsulates a designed DPModel to fuse the two kinds of features. FDPResnet first extracts the shallow and the deep scene features of a remote sensing image through a pre-trained model on residual neural network-101 (Resnet101). Then, it inputs the two kinds of features into a designed deep perception module (DPModel) to obtain a new set of feature vectors that can describe both higher-level semantic and lower-level space information of the image. The DPModel is the key module responsible for dimension reduction and feature fusion. Finally, the obtained new feature vector is input into BLS for training and classification, and we can obtain a satisfactory classification result. A series of experiments are conducted on the challenging NWPU-RESISC45 remote sensing image dataset, and the results demonstrate that our approach outperforms some popular state-of-the-art deep learning methods, and present high-accurate scene classification within a shorter running time.

Keywords: high-resolution remote sensing images; broad learning system (BLS); scene classification; ResNet101; feature representation

1. Introduction

The ever-advancing remote sensing technology now can generate a large number of high-resolution remote sensing images in a fast and effective way. This deftness has promoted its applications throughout numerous fields, including natural disaster monitoring, geospatial object detection, traffic supervision, weapon guidance and urban planning [1–3]. High-resolution remote sensing images, however, contain unique characteristics that make classifying scenes in them quite difficult. They usually stretch in different sizes, and contain diversified contents, like multi-directional targets standing against complex backgrounds. Some scenes in different classes may exhibit similar geographical features, while some belonging to the same class may look quite different, which easily leads scene classification to the wrong direction. Therefore, how to effectively classify scenes in high-resolution remote sensing images remains a challenging task.

The breakthrough that deep convolutional neural networks (DCNN) brought to image classification [4] has attracted attention from both academia and industry. DCNN exhibit the extraordinary ability of feature expression and outstrip traditional methods of image processing. They have been widely used throughout diverse research fields of computer vision, as well as target recognition and detection for natural images and other tasks [5,6]. A DCNN consists of a multi-stage architecture composed of convolution, pooling, and fully connected layers. At present, several powerful deep convolutional network models have been constructed for image processing, and representative network models are AlexNet [4], VGGNet [7], GoogLeNet [8], ResNet [9], etc.

Many researchers in remote sensing have noticed DCNN and adopted them for scene recognition of remote sensing images [10–14]. Actually, training a brand-new network specifically for high-resolution remote images is impractical, because it requires numerous samples and much longer training time. Plus, labeled samples are still insufficient, and training on small amount of samples would cause over-fitting. Current solution is to pretrain DCNN models on large datasets, like ImageNet [4], extract features from the trained models, and fuse those features to improve classification accuracy. Liu et al. [15] proposed a method to fuse all features of that DCNN convolutional layers provided and input them into a support vector machine (SVM) classifier. Penatti et al. [11] further improved classification accuracy by fusing the extracted CaffeNet and OverFeat features and inputting them into a SVM classifier [16]. Grant J. Scott [17] investigated a variety of fusion techniques and integrated multiple DCNN land cover classifiers into a single aggregate classifier. Their experiments reached satisfactory results of better accuracy and lower errors. [18] proposed a two-stage deep feature fusion model to improve the classification performance. The model can adaptively and explicitly combine the activations from intermediate and FC layers to generate a new CNN with directed acyclic graph topology, and achieve a state-of-the-art performance in scene classification tasks. Souleyman Chaib [19] designed a framework that targets VHR images. It establishes a VGG-Net-trained model to extract information from VHR images, and adopts, the discriminant correlation analysis (DCA) to fuse features selected from the fully connected layers. This approach can better describe the low-dimensional scenes in remote sensing images and achieve a higher recognition rate. These studies demonstrate that extracting features from convolutional neural networks and fusing them is an effective way to improve recognition rates. Most of them, however, adopt support vector machine (SVM) to classificate the extracted and fused features [11,15,17,19]. SVM projects samples into a high-dimensional space through a kernel function, and distinguishes sample categories by learning to classify hyperplanes. It can achieve satisfactory classification within a small amount of samples [20], but costs a longer training time for a large amount of samples. Scene recognition in an extensive dataset, like remote sensing image dataset NWPU-RESISC45 [21], requires a more efficient alternative.

Broad Learning System (BLS) [22], proposed by Chen and Liu, is an efficient and effective approximation approach for deep learning. BLS transfers the original inputs into “mapped features” and expands the structure with ”enhancement nodes”, thus the whole system is strengthened. This network requires no iteration and the output weights can be easily calculated by ridge regression. BLS has been conducted on some datasets, like Minist, and NORB, and effective performances have been fulfilled beyond the benchmark. However, it fails to achieve expected results on some large-scale remote sensing datasets, because the images there are large and have high dimensions. Some scholars have made certain improvements. Liu [23] proposed a modified broad learning structure based on the K-means feature extraction. The model takes both of their advantages and presents acceptable results, and this work proves that BLS model is flexible and has potential for various applications. Kong et al. [24] introduced BLS into the field of hyperspectral remote sensing image classification, and proposed a semi-supervised image learning method. Their experiments demonstrated the approach’s superiority.

Motivated by these advantages, we propose a fast deep perception network based on ResNet101 (Abbreviated as FDPResNet) that specifically targets scene classification for remote sensing images. First, a ResNet101 [9] model is trained on ImageNet to extract shallow and deep features of remote sensing images. Second, the shallow and the deep features are input into the deep perception module and converted to a set of depth-dense vectors. Finally, the vectors are input into the BLS-based pattern recognition system for training and classification. The contribution of our study can be summarized as follows,

1. Our study integrates DCNN and BLS into a framework to appropriate both DCNN's effective feature learning and BLS's fast decision-making. The proposed framework inherits can obtain the semantic information in high-resolution remote sensing images, as well as achieve fast pattern recognition.
2. We propose a deep perception model (DPModel)that can utilize both shallow and deep features of an image and extract richer semantic information from it. The model uses near-scale averaging operation to average the obtained shallow features, that is, integrating close convolutional layers into new convolutional layers that are then transformed into feature vectors through a vectorization operation. DPModel also adopts principal component analysis (PCA) [25] to avoid curse of dimensionality that arises with high dimensional vectors after features aggregation. Finally, the model cascades deep features and shallow features after dimension reduction from top to bottom and present new feature vectors that can present richer semantic information of the image.

The rest of the paper is organized as follows: Section 2 describes the principles and workflows of the proposed method; Section 3 presents the experiment and discussions. Conclusions are drawn in Section 4.

2. The Proposed Model

Figure 1 explains the framework of the proposed FDPResNet. It consists of three steps. First, the high-resolution remote sensing images are input into a model that is pre-trained by ImageNet on ResNet101, where no retraining or fine-tuning the network is involved, and the shallow and the deep features are obtained. Second, the shallow features are integrated by close-scale averaging operations, and the nearby convolutional layers are converged to new convolutional layers. The features the new layers contain are then flattened into vectors. Here PCA is adopted to reduce the dimensions of these new vectors, and thus the reduced shallow features and the deep features are cascaded to form new depth-dense feature vectors. Third, the depth-dense feature vectors are trained in the BLS network for classification.

2.1. Feature Extraction

Residual neural networks are a kind of deep neural networks that are based on the highway networks, proposed by He et al. [9]. The network replaces the gateway unit in the highway network with a shortcut connection to reduce network parameters while preserving original information. The advent of ResNet is a milestone of the advancement of deep learning. ResNet can accelerate the training of ultra-deep neural networks while greatly improving the accuracy. It can also circumvent the tricky situation that the increasing number of network layers would incur gradient disappearance or gradient explosion, which makes training extremely deep networks possible. Recently, some researchers have proved that ResNet with only one neuron per hidden layer is a general function approximator. The identity mapping enhances the expression ability of deep networks, and also indicates that Resnet can reduce the redundancy of information in data [26]. Therefore, we choose ResNet101 as the backbone network of our approach.

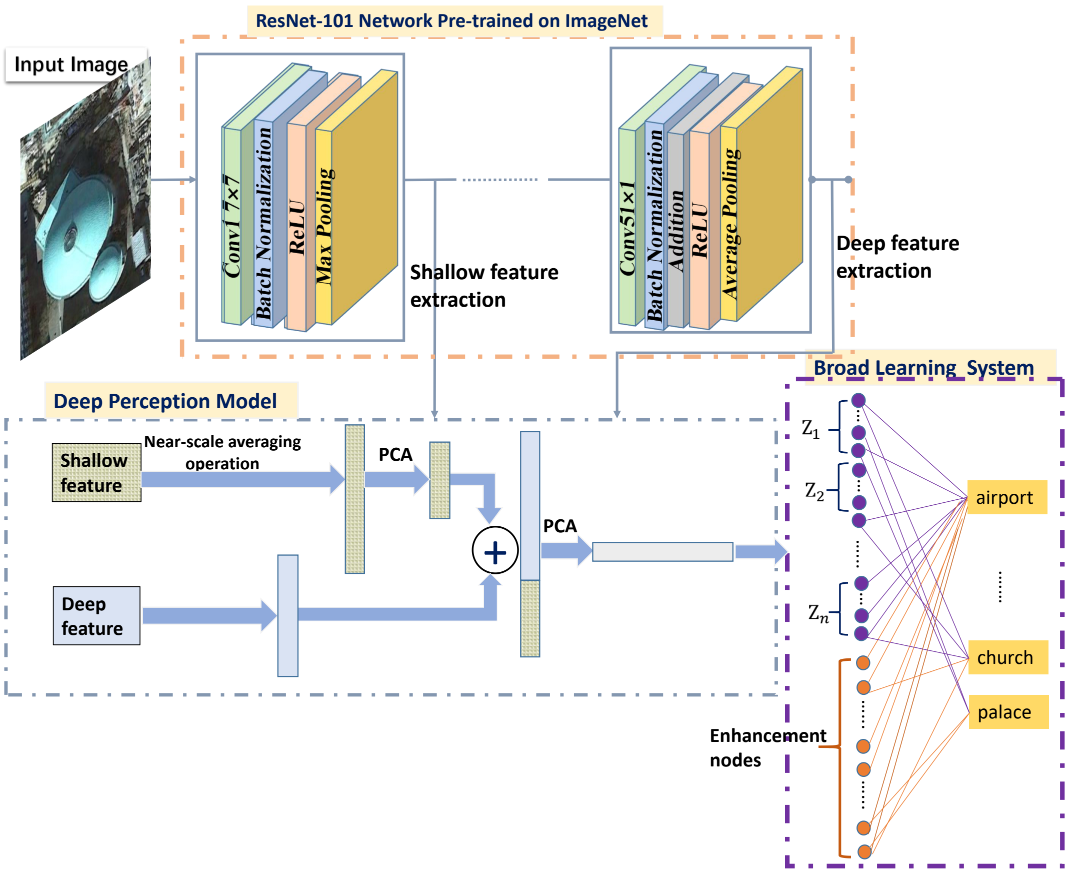


Figure 1. Framework of the proposed fast deep perception network (FDPResnet).

Current scene classification for remote sensing images usually extracts features from the convolutional layers of the pretrained CNN model or features of the most end (fully connected layers). The former contains local information and rich spatial data, while the latter contains semantic category information but lacks enough spatial information. Exploiting the features of the two kinds that mutually complement each other can provide a powerful representation. Therefore, FDPResNet extracts information from both the convolutional layer and the fully connected layer in this step.

This paper adopts a model that is trained by ResNet101 on ImageNet for feature extraction. The extracted features are divided into two categories: (1) The shallow features that have looped through the first convolution and the max pooling of the ResNet101 pre-trained model. (2) The deep features that have undergone the fifth convolution and the average pooling of the ResNet101 pre-trained model (close to the last layer of the classification performance).

The detailed process of extracting features is elaborated as follows. First, an image I that has been trimmed as $n \times n$ to fit training is input into the network. Second, The image loops through the network in a forward direction. Suppose there is a convolution layer L_l, locating in the lth layer. After the image passes the layer L_l, a $m \times m \times d$ feature map M^l can be obtained. For convenience, we denote $map_l = m \times m$. Thus for a feature map M^l, the feature of each frame can be denoted as map_l^i, where $1 \leq i \leq d$, d represents the overall dimension of the shallow feature, and the feature map M^l can be represented as $M^l = \left\{map_l^1, map_l^2, ..., map_l^d\right\} \in \mathbb{R}^{m \times m \times d}$. Therefore, the extracted shallow features can be denoted as:

$$M^1 = \left\{map_1^1, map_1^2, ..., map_1^{64}\right\} \in \mathbb{R}^{56 \times 56 \times 64} \tag{1}$$

The deep features can be expressed as:

$$M^5 = \left\{ map_5^1, map_5^2, \ldots, map_5^{2048} \right\} \in \mathbb{R}^{1\times1\times2048} \tag{2}$$

Third, M^1 and M^5 are input into the depth perception module to obtain the final features that can represent the image.

2.2. Deep Perception Model

Figure 2 is the internal process of the deep perception module (DPModel), which is divided into three steps.

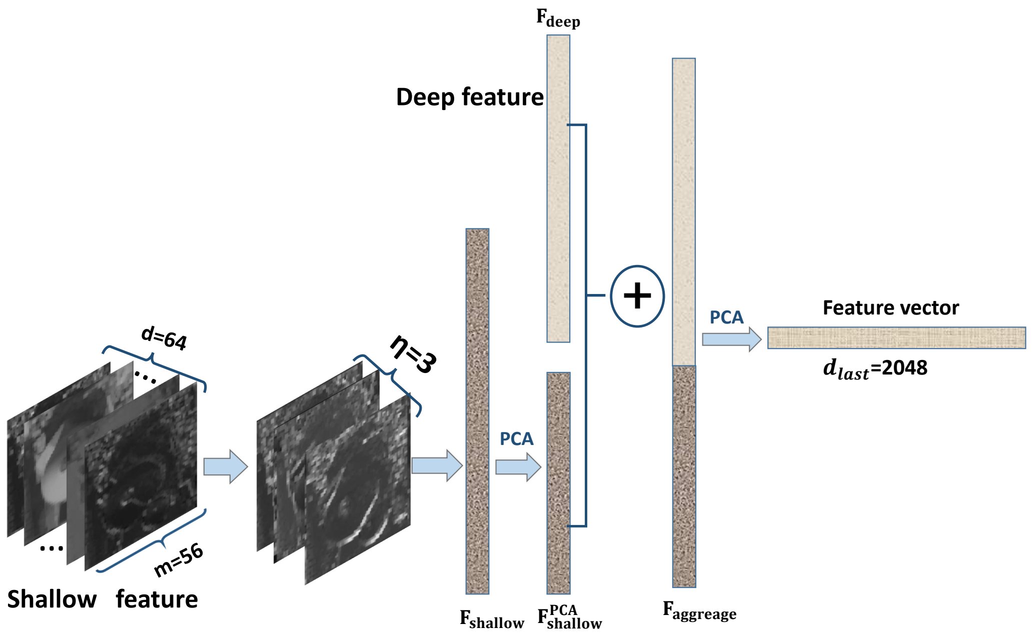


Figure 2. Deep perception module.

2.2.1. Processing Shallow Features

The obtained feature M^1 as described in Section 2.1 is a multi-dimensional feature. Figure 3 displays the visualization of M^1. It implies that not every channel feature can effectively represent the spatial information of the image. To this end, we propose a strategy of near-scale averaging to intelligently extract some shallow features from this d dimensional feature map, thus the features can represent texture information and spatial direction information.

As mentioned in Section 2.1, an $m \times m$ feature map can be denoted as map_l^i. Suppose $F_{x,y}^i$ is a single eigenvalue in map_l^i, where $1 \le x \le m$,$1 \le y \le m$, and $\overline{F_{ave}^l}$ represents mean of the eigenvalues of map_l^i.

$$\overline{F_{ave}^l} = \frac{1}{m \times m} \sum_{x=1,y=1}^{x=m,y=m} F_{x,y}^i \tag{3}$$

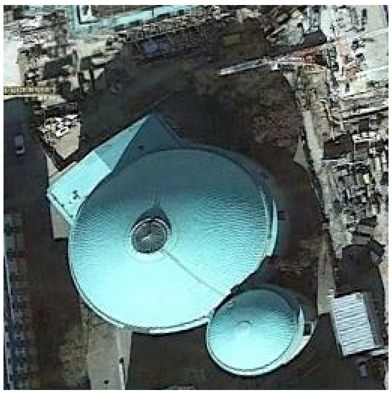
For M^1, the set of mean eigenvalues of each channel can be expressed as $\overline{F^1} = \{\overline{F^1_{ave}}, \overline{F^2_{ave}}, \ldots, \overline{F^d_{ave}}\}$. Then the sum of values in $\overline{F^1}$ set, denoted as MAP can be expressed as

$$MAP = \frac{1}{d}\sum_{i=1}^{i=d}\overline{F^l_{ave}} \tag{4}$$

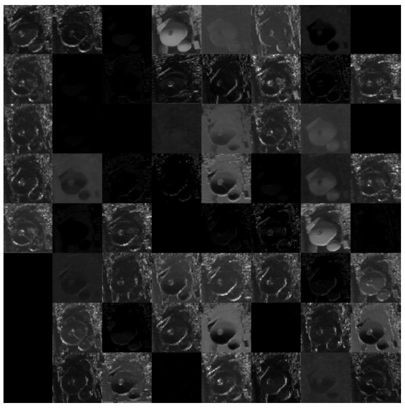
We need to find such a channel in M^1:

$$|MAP - F^1_{ave}| \in [0, 0.5] \tag{5}$$

Thus, the channel that is the closest to the average can be obtained as $C = \{c_1, c_2, \ldots, c_\eta\}$, and the feature set of the channel can be denoted as $MAP_{aim} = \{MAP_1^{c1}, MAP_1^{c2}, MAP_1^{c3}, \ldots, MAP_1^{c\eta}\} \in \mathbb{R}^{56\times56\times\eta}$, let $Dim = m \times m \times \eta$. The results of several experiments we have conducted demonstrate that performance of scene recognition will reach the best when $\eta = 3$, Therefore, MAP_{aim} can be flattened into a vector of $1 \times Dim$, denoted as $\mathbf{F_{shallow}}$.



(**a**)



(**b**)

Figure 3. Visualization of shallow features in a remote sensing image. (**a**) Original. (**b**) Visualization of shallow features.

2.2.2. PCA Reducing Dimensions

The dimensions of $\mathbf{F_{shallow}}$ have been reduced to η after the previous step, but still contain redundant features. For example, suppose $\mathbf{F_{shallow}} = 1 \times (m \times m \times \eta)$. When $m = 54$, $\eta = 3$, the dimensions of the shallow feature is 1×8748, which will easily cause curse of dimensionality. Therefore, we adopt principal component analysis (PCA) to reduce the dimension of $\mathbf{F_{shallow}}$.

The core of PCA is to project each sample to the sample space in a direction toward large variance, that is, to find a large variance component in order to retain the original information as much as possible after dimension reduction. In other words, PCA avoids curse of dimensionality through discarding part of the information during dimension reduction. The features after PCA dimension reduction are also independent from each other [27].

In this paper, the reduced dimension is denoted as $D^{PCA}_{shallow}$. Given the key role that an appropriate dimension could play in classification, we conducted a grid search between $[256, 2048]$ with an nterval of 128 to find a perfect dimension, and find that when $D^{PCA}_{shallow} = 512$, the obtained shallow features achieve better scene classification. Thus, the feature after dimension reduction is denoted as $\mathbf{F^{PCA}_{shallow}}$.

2.2.3. Aggregation of Features

The deep features obtained through feature extraction, as described in Section 2.1, represent the high-level semantic information and the shallow features describe the space information and texture information. Scene classification mostly depends on the semantic information, and uses the space and texture information as supplement. Therefore, our approach adopts a top-down aggregation to fuse the two kinds of features. First, the deep feature M^5 obtained through feature extraction is flattened into a vector $\mathbf{F_{deep}}$ with a dimension $1 \times d_{last}$; Then, the two kinds of features can be aggregated as $\mathbf{F_{aggreage}} = \left[\mathbf{F_{deep}} \quad \mathbf{F^{PCA}_{shallow}}\right]$, where the dimension of $\mathbf{F_{aggreage}}$ is $1 \times (d_{last} + D^{PCA}_{shallow})$. PCA is also utilized here to find the optimal dimension, and results indicated that when dimension of $\mathbf{F_{deep}}$, d_{last} is 2048, the best classification results can be achieved.

2.3. Broad Learning System

Broad learning system is actually a derivative variant of the random vector functional link neural network (RVFLNN) [28]. A BLS network [22], as exemplified as the purple box in Figure 1, works in three steps: First, the features of the input data mapping are used as the "feature nodes" of the network; second, the features of the mapping are elevated to "enhanced nodes" with randomly generated weights; third, all mapped features and enhanced nodes are directly connected to the output, and the corresponding output coefficients can be derived from the fast pseudo-inverse.

According to Section 2.2.3, the depth-dense vector is denoted $\mathbf{F_{aggreage}}$ with a dimension D of 2048. Suppose the number of samples in the data is N, the input sample can be defined ad $\mathbf{F} \in \mathbb{R}^{\mathbf{N \times D}}$, and $\mathbf{F} = \left[\mathbf{F^1_{aggreage}}, \mathbf{F^2_{aggreage}}, \ldots, \mathbf{F^N_{aggreage}}\right]$. The mapping feature in the BLS system is set to $\mathbf{Z}$ and the number of feature nodes are b, then the mapping feature of the dataset on the feature plane is

$$\mathbf{Z}^{N \times b} = \mathbf{F}^{N \times D} \cdot \mathbf{W}_e^{D \times b} \tag{6}$$

where W_e is the optimal input weight matrix obtained by sparse self-encoding. If k enhanced nodes are generated in BLS, and $\mathbf{H}$ is used to represent the enhanced feature matrix, the enhanced feature matrix of the dataset can be expressed as,

$$\mathbf{H}^{N \times k} = \phi(\mathbf{Z}^{N \times b} \cdot \mathbf{W}_h^{b \times k} + \beta_h^{N \times k}) \tag{7}$$

where $\mathbf{W}_h$ and β_h represent the random matrix and bias respectively; $\phi(\cdot)$ is an optional non-linear activation function, and *tansig* is selected as the excitation function in this paper. BLS is a merging matrix that connects feature nodes and enhancement nodes. The merging matrix is the actual input of BLS, that is, $\mathbf{A} = \left[Z^{N \times b} | H^{N \times k} \right]$. We assume that the output matrix is $\mathbf{L} \in \mathbb{R}^{N \times C}$, where C represents the number of categories of the dataset, then the output matrix can be obtained according to BLS as,

$$\mathbf{L}^{N \times C} = \mathbf{A}^{N \times (b+k)} \cdot \mathbf{W}^{(b+k) \times C} = \left[\mathbf{Z}^{N \times b} | \mathbf{H}^{N \times k} \right] \cdot \mathbf{W}^{(b+k) \times C} \tag{8}$$

where $\mathbf{W}$ represents the connection weight matrix, and $\mathbf{W}$ is obtained by taking the ridge regression approximation of $\mathbf{A}^{+}$, as shown in Equation (9)

$$\mathbf{A}^{+} = \lim_{\lambda \to 0} \left(\lambda \mathbf{I} + \mathbf{A}\mathbf{A}^{T} \right)^{-1} \mathbf{A}^{T} \tag{9}$$

Therefore, in a BLS system, the network only needs to learn the output matrix $\mathbf{W}$. In this formula, λ is a regular l2-norm regularization, and set as $\lambda = 2^{-10}$.

3. Experiments and Results

3.1. Dataset

We conducted a series of experiments on NWPU-RESISC45 dataset [21], which was created by a research team of the Northwestern Polytechnic University in 2017. It contains 31,500 remote sensing images and 45 scene categories. Each scene category contains 700 images of size 256×256. The spatial resolution of most images can reach 30 m$\sim$0.2 m/pixel, and images in certain categories of special landforms may be in lower-resolution, like islands, lakes, regular mountains and snow mountains. This dataset includes abundant scene categories, and each category retains enough inner-diversity and inter-similarity with other classes. It is a challenging benchmark to test scene classification for remote sensing images.

3.2. Implementation Details

We randomly selected training samples and test samples from each category at a ratio of 2:8 and 1:9, respectively. Each set of experiments was repeated 10 times. The final classification performance was evaluated by the average of the accuracies of all experiments. We adopted the LibLinear library [16] to exert linear SVM training and testing.

All the experiments were conducted on a personal computer with a quad-core CPU of 2, 4 GHz, a graphics card of GeForece GT1080 8G GPU, and equipped with MathWorks MATLAB R2018b. Multiple pre-tests helped us to determine the values of key variables: $\eta = 3$, $D_{shallow}^{PCA} = 512$, $\lambda = 2^{-10}$.

3.3. Effectiveness of Fusion of Shallow and Deep Features

We used ResNet101 to extract shallow features and deep features from images in NWPU-RESISC45, and used our approach to fuse the two levels of features. Then, three kinds of results were visualized by t-SNE and exhibited in Figure 4. Each dot in Figure 4 represents a sample of a category, and different categories are distinguished by different colors. The shallow features in Figure 4a are nearly messy lines of dots that demonstrate linear features, rather than clustering characteristics; while the deep features in Figure 4b are presented in a high level of abstraction with obvious clustering features, but the boundaries between classes are still blurred. The fusion of the two levels of features in Figure 4c indicates the final features of those categories. Compared to Figure 4a–c presents a clearer classification with increased distances and similarity between classes. The final results validate that our approach of feature fusion can effectively improve scene classification performance.

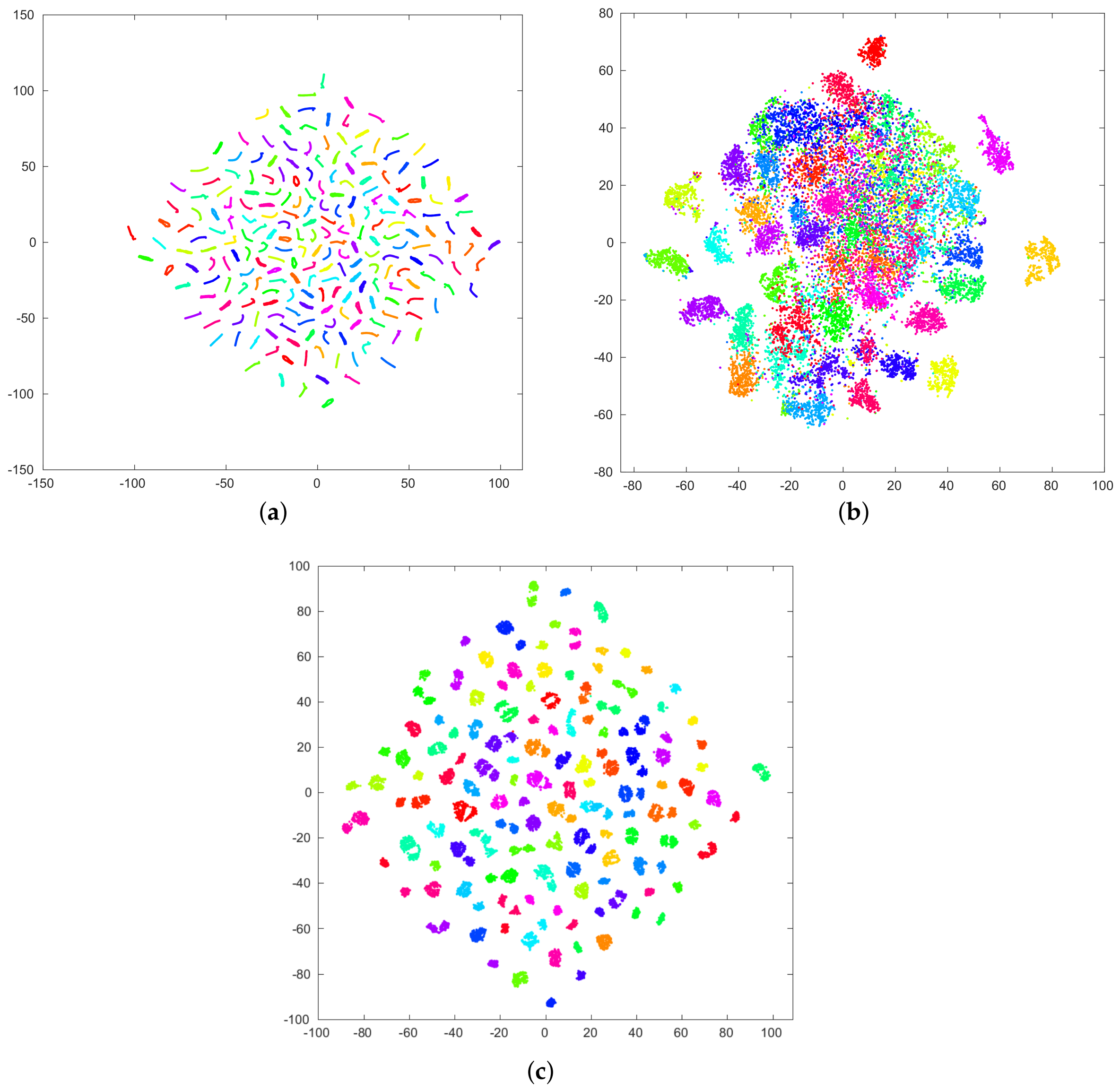


Figure 4. Two-dimensional visualization results of scene features for images in NWPU-RESISC45 dataset. In this paper, t-SNE was used to map shallow features, high-level features, and shallow and high-level features into 2D space. (**a**) Shallow features, (**b**), High-level features, (**c**) Fusion of shallow and high-level features.

3.4. *Comparison of FDPResNet and Other State-of-the-Art Methods*

3.4.1. Comparison of the Accuracy

We compared the classification results of the proposed FDPResNet with those of seven state-of-the-art methods that were also conducted on NWPU-RESISC45 [21], as elaborated in Table 1. The accuracies in Table 1 indicate that our approach is superior to both transferred learning and fine-tuning methods [21]. Compared with the result of the D-CNN with VGGNet-16 [13] method that performed the best among the seven, the accuracy of our approach is 4% higher. The accuracy comparison demonstrates better effectiveness of our approach on NWPU-RESISC45.

Table 1. Comparison of accuracies and standard deviations (%) of different methods on the NWPU-RESISC45 dataset.

Method	Training Ratio	
	20%	10%
Transferred AlexNet + SVM [21]	79.85 ± 0.13	76.69 ± 0.21
Transferred GoogLeNet + SVM [21]	78.48 ± 0.26	76.19 ± 0.38
Transferred VGGNet-16 + SVM [21]	79.79 ± 0.15	76.47 ± 0.18
Fine-tuned AlexNet + SVM [21]	85.16 ± 0.18	81.22 ± 0.19
Fine-tuned GoogLeNet + SVM [21]	86.02 ± 0.18	82.57 ± 0.12
Fine-tuned VGGNet-16 + SVM [21]	90.36 ± 0.18	82.15 ± 0.45
D-CNN with VGGNet-16 [13]	91.89 ± 0.22	89.22 ± 0.50
FDPResNet	**95.40 ± 0.11**	**92.32 ± 0.32**

3.4.2. Comparison of Training and Testing Time

Table 2 compares the training time and test time of the proposed FDPResNet and those methods [21] with SVM as the classifier. Here, the ratio of the training samples and test samples on the NWPU-RESISC45 dataset is 20%. Table 2 indicates that compared with the methods using SVM [21], FDPResNet saves approximately three times the training time and four times the test time. We also input the vectors obtained from our proposed DPModel into a SVM-based method for classification, and the resulted accuracy is higher than that of the six methods [21] and lower than that of the FDPResNet framework. This well demonstrates that features obtained by the DPModel can better represent the semantic information of the image, and has good generalization. As for operating efficiency, the training time and the test time that the DRResNet+SVM methods cost are greater than those of the FDPResNet, which means that BLS adopted in the FDPResNet framework can better fulfill scene classification for remote sensing images.

Table 2. Comparison of training time and testing time of different methods on the NWPU-RESISC45 dataset.

Method	Overall Accuracies (20%)	Training Times (s)	Testing Times (s)
Transferred AlexNet + SVM	79.85 ± 0.13	163.23	69.12
Transferred GoogLeNet + SVM	78.48 ± 0.26	238.46	79.23
Transferred VGGNet-16 + SVM	79.79 ± 0.15	196.11	73.24
Fine-tuned AlexNet + SVM	85.16 ± 0.18	165.25	68.86
Fine-tuned GoogLeNet + SVM	86.02 ± 0.18	235.46	78.46
Fine-tuned VGGNet-16 + SVM	90.36 ± 0.18	195.25	72.42
DPModel + SVM	91.8 9 ± 0.22	162.05	75.41
FDPResNet	**95.40 ± 0.11**	**62.8526**	**17.2844**

3.4.3. Confusion Matrix

To better understand the performance of our approach, we depicted a confusion matrix to illustrate the correctness of the classification results, as shown in Figure 5. Each row of the matrix represents the class predicted by our approach while each column represents the actual class. Thus, cells on the diagonal indicate the correct prediction while cells on other are as imply errors. Numbers in each cell represent the total number and the percentage of the predicted instances, and the classes are organized in a descending order of correctness along the diagonal from the left to the right.

In Figure 5, the recognition accuracies higher than 90% appear in more than 86% of the classes, and those lower than 80% only account for 4% of the classes. This further demonstrates that the FDPResNet is applicable to images with complex contents, like those in NWPU-RESISC45. Those scenes that the FDPResNet can recognize in high accuracy contain single texture and exhibit little within-class similarity, such as *basketball_court*, *rectangular_farmland*, *mountain*; and those that the FDPRseNet recognized in low accuracy contain more complex contents, exhibit high similarities between-class and larger diversity within-class, such as *wetland*, *commercial_area*, *intersection*. While great success has been obtained so far, the problems of within-class diversity and between-class similarity are still two big challenges.

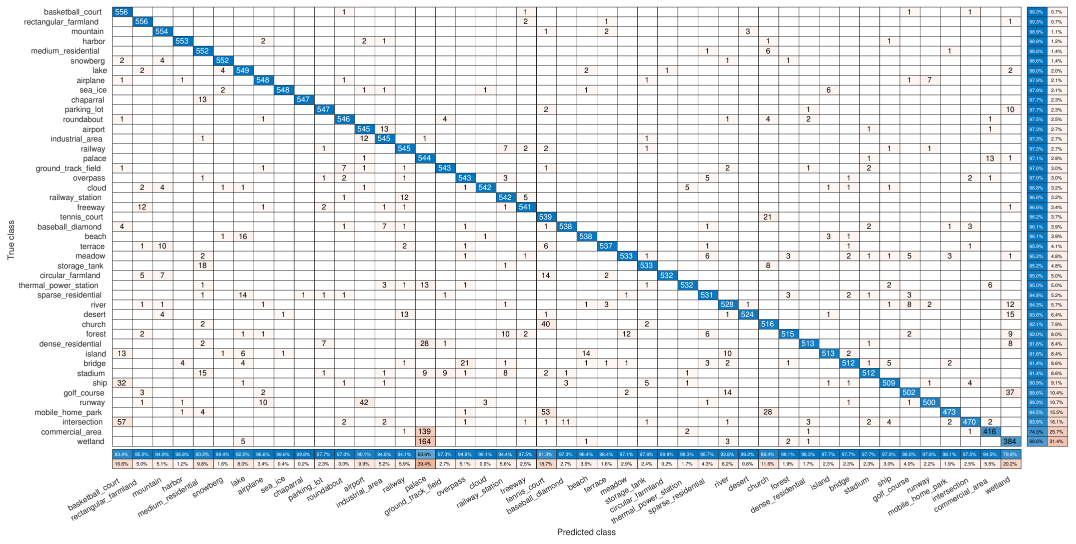


Figure 5. Confusion matrix of the proposed FDPResNet on NWPU-RESISC45 dataset.

4. Conclusions

This paper proposes a novel fast deep perception network (FDPResNet) to utilize the expertise of DCNN and BLS. The FDPResNet uses a model pre-trained by ImageNet on the ResNet101 to extract both shallow features and deep features from an image, and then inputs these two kinds of features into the proposed DPModel to obtain a set of depth-dense vectors that can represent semantic information of the image. Consequently, BLS can utilize this set of deep dense vectors and outputs satisfactory scene classification results. The comparison experiments on the challenging dataset NWPU-RESISC45 [21] demonstrates that the FPDResNet can achieve optimal performance. Future work will focus on improving the FDPResNet's classification accuracy for scenes that are ambiguous.

Author Contributions: All of the authors made significant contributions to the work. Conceptualization, L.J. and J.Z.; methodology, D.X. and J.Z.; software, R.D.; validation, R.D., J.Z. and J.A.; writing—original draft preparation, R.D.; writing—review and editing, R.D. and D.X.; supervision, J.Z.; funding acquisition, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China grant No. 61471192, 61371169, and the Priority Academic Program Development of Jiangsu Higher Education Institutions, and Foundation of Graduate Innovation Center in NUAA grant No. kfjj20160414.

Acknowledgments: The author would like to thank Chen Junlong for the source code, and the anonymous reviewers for their constructive and valuable suggestions on the earlier drafts of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, S.; Fang, T.; Li, D.; Wang, S. Object Classification of Aerial Images With Bag-of-Visual Words. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 366–370.

2. Huang, X.; Liu, H.; Zhang, L. Spatiotemporal Detection and Analysis of Urban Villages in Mega City Regions of China Using High-Resolution Remotely Sensed Imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3639–3657. [CrossRef]
3. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS'12, Lake Tahoe, NV, USA, 3–8 December 2012; Curran Associates Inc.: Red Hook, NY, USA, 2012; Volume 1, pp. 1097–1105.
5. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst*. **2014**; *27*, 3320–3328.
6. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 512–519.
7. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:abs/1409.1556.
8. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
10. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning Low Dimensional Convolutional Neural Networks for High-Resolution Remote Sensing Image Retrieval. *Remote Sens.* **2016**, *9*, 489. [CrossRef]
11. Penatti, O.A.; Nogueira, K.; Dos Santos, J.A. 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, Boston, MA, USA, 7–12 June 2015; pp. 44–51.
12. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
13. Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2811–2821. [CrossRef]
14. Dong, R.; Xu, D.; Zhao, J.; Jiao, L.; An, J. Sig-NMS-Based Faster R-CNN Combining Transfer Learning for Small Target Detection in VHR Optical Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8534–8545. [CrossRef]
15. Liu, Q.; Hang, R.; Song, H.; Zhu, F.; Plaza, J.; Plaza, A. Adaptive Deep Pyramid Matching for Remote Sensing Scene Classification. *arXiv* **2016**, arXiv:1611.03589.
16. Chang, C.C.; Lin, C.J. LIBSVM: A Library for Support Vector Machines. 2001. Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed on 7 May 2018).
17. Scott, G.J.; Hagan, K.C.; Marcum, R.A.; Hurt, J.A.; Anderson, D.T.; Davis, C.H. Enhanced Fusion of Deep Neural Networks for Classification of Benchmark High-Resolution Image Data Sets. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1451–1455. [CrossRef]
18. Liu, Y.; Liu, Y.; Ding, L. Scene Classification Based on Two-Stage Deep Feature Fusion. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 183–186. [CrossRef]
19. Chaib, S.; Liu, H.; Gu, Y.; Yao, H. Deep Feature Fusion for VHR Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4775–4784. [CrossRef]
20. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
21. Gong, C.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883.
22. Chen, C.; Liu, Z. Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 10–24. [CrossRef] [PubMed]

23. Liu, Z.; Zhou, J.; Chen, C.P. Broad learning system: Feature extraction based on K-means clustering algorithm. In Proceedings of the IEEE 4th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Dalian, China, 24–26 July 2017; pp. 683–687.
24. Kong, Y.; Wang, X.; Cheng, Y.; Chen, C. Hyperspectral imagery classification based on semi-supervised broad learning system. *Remote Sens.* **2018**, *10*, 685. [CrossRef]
25. Agarwal, A.; El-Ghazawi, T.; El-Askary, H.; Le-Moigne, J. Efficient Hierarchical-PCA Dimension Reduction for Hyperspectral Imagery. In Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, Cairo, Egypt, 15–18 December 2007; pp. 353–356.
26. Lin, H.; Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. *arXiv* **2018**, arXiv:1806.10909v2.
27. Cao, L.; Chua, K.S.; Chong, W.K.; Lee, H.P.; Gu, Q.M. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing* **2003**, *55*, 321–336. [CrossRef]
28. Pao, Y.H.; Takefuji, Y. Functional-link net computing: theory, system architecture, and functionalities. *IEEE Comput.* **1992**, *25*, 76–79. [CrossRef]