International Journal of
Geo-Information

MDPI

Article

Improving the Separability of Deep Features with Discriminative Convolution Filters for RSI Classification

Na Liu [1,2,*], Xiankai Lu [1,2], Lihong Wan [1,2], Hong Huo [1,2] and Tao Fang [1,2,*]

[1] Institute of Image Processing and Pattern Recognition, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; carrierlxk@sjtu.edu.cn (X.L.); wan611@sjtu.edu.cn (L.W.); huohong@sjtu.edu.cn (H.H.)

[2] Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, China

* Correspondence: a8152692@sjtu.edu.cn (N.L.); tfang@sjtu.edu.cn (T.F.)

Received: 31 December 2017; Accepted: 7 March 2018; Published: 12 March 2018

Abstract: The extraction of activation vectors (or deep features) from the fully connected layers of a convolutional neural network (CNN) model is widely used for remote sensing image (RSI) representation. In this study, we propose to learn discriminative convolution filter (DCF) based on class-specific separability criteria for linear transformation of deep features. In particular, two types of pretrained CNN called CaffeNet and VGG-VD16 are introduced to illustrate the generality of the proposed DCF. The activation vectors extracted from the fully connected layers of a CNN are rearranged into the form of an image matrix, from which a spatial arrangement of local patches is extracted using sliding window strategy. DCF learning is then performed on each local patch individually to obtain the corresponding discriminative convolution kernel through generalized eigenvalue decomposition. The proposed DCF learning characterizes that a convolutional kernel with small size (e.g., 3 × 3 pixels) can be effectively learned on a small-size local patch (e.g., 8 × 8 pixels), thereby ensuring that the linear transformation of deep features can maintain low computational complexity. Experiments on two RSI datasets demonstrate the effectiveness of DCF in improving the classification performances of deep features without increasing dimensionality.

Keywords: convolutional neural network; discriminative convolution filter; image classification; remote sensing

1. Introduction

In recent years, remote sensing image (RSI) classification has attracted remarkable attention and is becoming increasingly important in a wide range of applications, such as geographic image retrieval, object detection, environment monitoring, and vegetation mapping [1]. Learning robust RSI representations plays an important role in RSI classification because rich geometric structures and spatial patterns exist in RSIs [2]. Over the past few years, numerous methods have been proposed for RSI classification. The methods can generally be categorized into three aspects according to feature type [1]: (1) methods based on handcrafted features; (2) methods based on unsupervised feature learning; and (3) methods based on deep learning.

For the first category, representative handcrafted features are color histogram [3], scale-invariant feature transform (SIFT) [4], histograms of oriented gradients (HOG) [5], local binary patterns (LBP) [6,7], Gabor [8], and GIST [9,10] which is an abstract representation of scene. Yang and Newsam [3] investigated color histogram and SIFT-based bag-of-visual-words (BOW) representations for RSI classification. Cheng et al. [5] proposed to train part detectors by using HOG feature pyramids to extract distinguishable features for RSI representation. Ren et al. [7] proposed LBP structure learning based on incremental maximal conditional mutual information. Risojevic and Babic [8] presented an enhanced Gabor texture descriptor (EGTD) based on cross-correlation within the spatial frequency subbands of Gabor decomposition. In general, methods based on handcrafted features usually extract single-feature cues from images and should be redesigned (e.g., parameter setting) for a new dataset, thereby making RSI classification heavily dependent on expert experiences.

For the second category, k-means clustering, sparse coding, and autoencoder are commonly used in unsupervised feature learning methods. As a vector quantization method, k-means clustering aims to partition training samples into k clusters, ensuring that the distances of the training samples within a cluster are similar while the training samples that belong to two different clusters are dissimilar. With the use of numerous local descriptors (e.g., SIFT, HOG, and LBP) extracted from a set of training images, k-means is widely used for learning the visual dictionary of BOW-based mid-level representation. Sparse coding [11] aims to learn an overcomplete dictionary from unlabeled samples, ensuring that an image can be efficiently represented through a linear combination of the basis functions in the overcomplete dictionary. Cheriyadat [12] proposed the use of sparse coding to learn a set of basis functions from low-level features (e.g., dense SIFT) for RSI classification. As a neural network method, autoencoder aims to learn a low-dimensional feature representation from high-dimensional features in an unsupervised manner. Zhou et al. [13] proposed to learn sparse features based on autoencoder for RSI retrieval. Othman et al. [14] proposed to use convolutional features and sparse autoencoder for RSI classification. Unlike handcrafted features, unsupervised feature learning can automatically learn meaningful features rather than find the best design of a given dataset.

For the third category, deep learning [15] has been intensively used for visual recognition, with convolutional neural network (CNN) [16] being a popular topic in the deep learning community for automatic learning of visual features. In the past years, numerous CNN architectures, such as AlexNet [16], VGG (Visual Geometry Group)-VD (Very Deep) [17], GoogLeNet [18], ResNet [19] have been proposed for image recognition (e.g., ImageNet dataset [20]). In the field of RSI classification, Zhang et al. [21] proposed a gradient-boosting random convolutional network (GBRCN) framework with the use of RSI training data for land use classification. However, fully designing and training a new CNN architecture in remote sensing applications is always difficult because training a CNN model requires a large-scale labeled dataset, which is unusual in the remote sensing community [22]. Therefore, determining how existing pretrained CNNs (e.g., trained on ImageNet) can be better used is an interesting task for obtaining high performances. Zeiler and Fergus [23] pointed out that the activation vector extracted from the fully connected layer of a CNN can be used as a powerful image descriptor for the feature extraction of other datasets. Penatti et al. [24] evaluated the generalization of deep features using pretrained CNN models such as CaffeNet [25] and OverFeat [26]. Castelluccio et al. [27] demonstrated that fine-tuned CNN could obtain better classification performance compared with pretrained CNN. Wan et al. [28] proposed a cascade representation framework for RSI classification based on a set of pretrained CNNs. Recently, Nogueira et al. [22] presented a comprehensive analysis of three possible strategies to investigate the power of existing CNNs. Their results illustrate that using the activation vectors extracted from the fully connected layer of a CNN model as RSI representations followed by linear support vector machine (SVM) [29] can yield the best classification performance.

In general, effective feature representation is beneficial for the subsequent stage of classifier training. In previous years, Kumar et al. [30,31] proposed the use of Volterra theory for the first time to learn discriminative convolution filters (DCF) from pixel features on gray-level images. The supervised learning process of DCF is based on a class-specific separability criteria, which can be converted into generalized eigenvalue decomposition. However, the pixel features within an image are strongly correlated to nearby pixels. With the development of deep learning, the deep features extracted by a CNN model can obtain different levels of data abstraction from pixel features. Deep features have less feature redundancy and higher distinguishability than pixel features, thereby making DCF learning on deep features more interesting than on pixel features.

According to the recent review of CNN [22], the best performing deep features for image representation (e.g., RSI) can be obtained from the fully connected layer. Deep features are represented by an activation vector rather than a feature map matrix because of the characteristics of fully connected layers. For example, several widely used CNN architectures, such as AlexNet, CaffeNet, and VGG-VD, contain 4096-dimensional activation vectors in their fully connected layers. DCF was originally proposed to learn spatial kernels on gray-level images with 64 × 64 pixels. Thus, vector-based deep features (activation vectors) cannot be directly used to learn spatial kernels through DCF. Interestingly, a 4096-dimensional activation vector can be reshaped to a matrix with 64 × 64 features through rearrangement, thereby making DCF learning on activation vectors feasible.

On this basis, we explore the first-order (linear) form of DCF learning for the transformation of activation vectors in this study to improve the separability of deep features and the classification performance of RSI without increasing the dimensionality of feature representation. Applying DCF learning to activation vectors is based on the hypothesis that the rearrangement of activation vectors contains discriminant spatial structures, which help increase the probability that a statistical learning model will reveal interesting regularities. In particular, the activation vectors in a CNN model are initially extracted from the fully connected layers. To illustrate the generality of DCF learning for activation vectors, two types of pretrained CNNs, namely, CaffeNet [25] and VGG-VD16 [17], are used for comprehensive performance comparisons. Then, the activation vector is rearranged into the form of an image matrix, from which local patches are extracted using the sliding window strategy. Finally, DCF learning is performed on each local patch individually through a generalized eigenvalue problem. The advantage of the proposed method is that small DCF kernels (e.g., 3 × 3 pixels) can be effectively learned on small local patches (e.g., 8 × 8 pixels). Therefore, the transformation (linear convolution) of the activation vectors can maintain low computational complexity. The effectiveness of DCF transformation for activation vectors is further evaluated by supervised classification with linear SVM, which is widely used in CNN-based RSI classification [1,2,22,24,28]. Experiments on two publicly available RSI datasets demonstrate that DCF helps improve the classification performance of the activation vectors obtained by CaffeNet or VGG-VD16.

2. Proposed Method

As shown in Figure 1, the proposed method consists of the following stages: (1) extracting 4096-dimensional activation vector from the fully connected layer of a pretrained CNN, followed by L2 normalization; (2) rearranging the activation vector into the form of an image matrix, from which small-size local patches are extracted by using the sliding window strategy with a fixed stride, followed by DCF learning on each local patch to obtain a convolutional filter; and (3) generating linear transformation of the deep features on the basis of the learned filters.

2.1. Deep Features

Pretrained CaffeNet and VGG-VD (obtained by MatConvNet [32]) are selected to extract deep features and to demonstrate the generality of the proposed DCF for the transformation of different types of deep feature. We select pretrained CNN to extract deep features because a CNN model learned on a large-scale dataset has good generalization on other tasks (e.g., RSI classification) and no retraining process is needed for the target application. Previous works [22,24] indicated that the deep features, which are the so-called activation vectors in this study, can be obtained from the last fully connected layers (except for the classification layer). Thus, such strategy is employed in our deep feature extraction.

2.1.1. CaffeNet

In 2012, Krizhevsky et al. [16] proposed a deep CNN architecture, which is the so-called AlexNet, for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and has gained great success in the ILSVRC 2012 competition. AlexNet is a breakthrough in image recognition because (1) nonsaturating neurons are used, (2) dropout technique is introduced to prevent overfitting, and (3) GPU implementation is used to accelerate the learning speed. As a reference model in the Caffe open source framework [25], CaffeNet is nearly a replication of AlexNet. Unlike AlexNet, CaffeNet has no data argumentation in the training stage, and the order of normalization and pooling operations in the CNN architecture is exchanged.

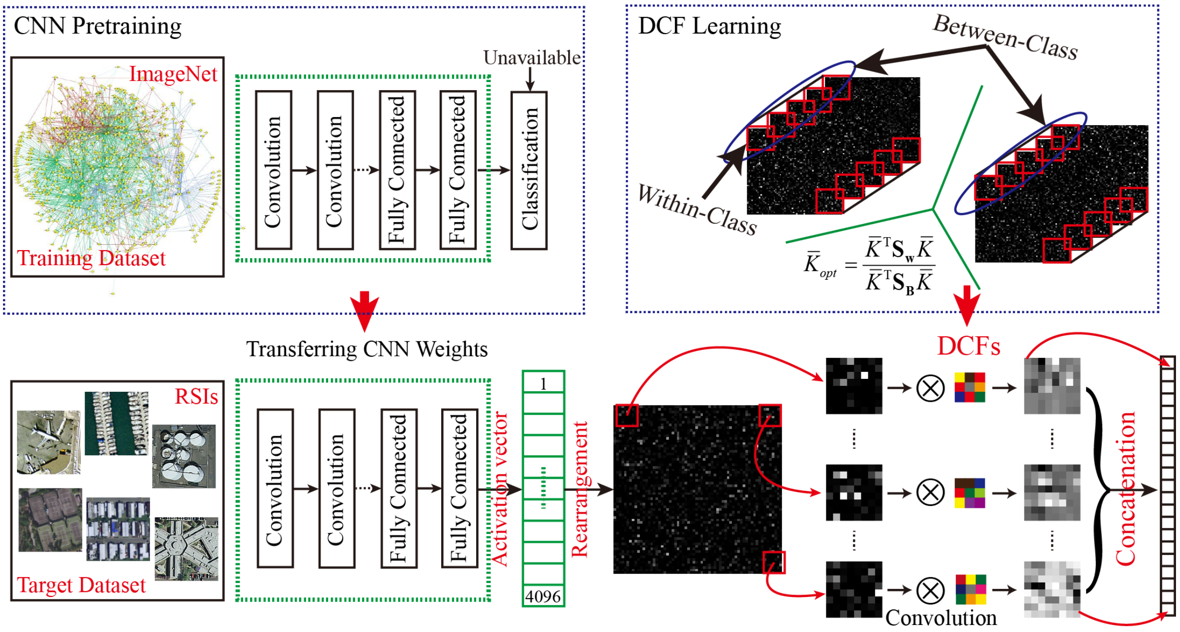


Figure 1. Framework of the proposed method based on deep features and DCF (Discriminative Convolution Filter). The top row contains the learning stages of pretrained CNN (Convolutional Neural Network) and DCF. The bottom explains the extraction of deep features and their convolutional transformation based on DCF kernels.

CaffeNet consists of five convolutional layers and three fully connected layers. The convolutional layers contain a linear convolution followed by one or more nonlinear operations, such as rectified linear units (ReLU), local response normalization, and max pooling. The input image size is 227 × 227 pixels with three channels (red-green-blue). The first convolutional layer contains 96 kernels (receptive fields or filters) with a size of 11 × 11 × 3 pixels. The second convolutional layer contains 256 kernels with a size of 5 × 5 × 48 pixels. The third convolutional layer contains 384 kernels with a size of 3 × 3 × 256 pixels. The fourth convolutional layer contains 384 kernels with a size of 3 × 3 × 192 pixels. The fifth convolutional layer contains 256 kernels with a size of 3 × 3 × 192 pixels. Each fully connected layer (except for the last classification layer) contains 4096 neurons. In this study, we extract 4096-dimensional activation vectors from the first and the second fully connected layers.

In addition, data augmentation shown in Figure 2 is performed by sampling sub-images from the original input image and averaging the activation vectors of these sub-images, similar to the prevalent "center + corners with horizontal flips" augmentation [16,28,33]. First, the original input image is resized to 256 × 256 pixels. Then, five sub-images (corresponding to the center and four corners) and their horizontal flips are cropped from the original image. Finally, each sub-image is used to extract two 4096-dimensional activation vectors from two fully connected layers. The final representation of the original input image can be obtained by averaging the 4096-dimensional activation vectors over the 20 sub-images, followed by L2 normalization.

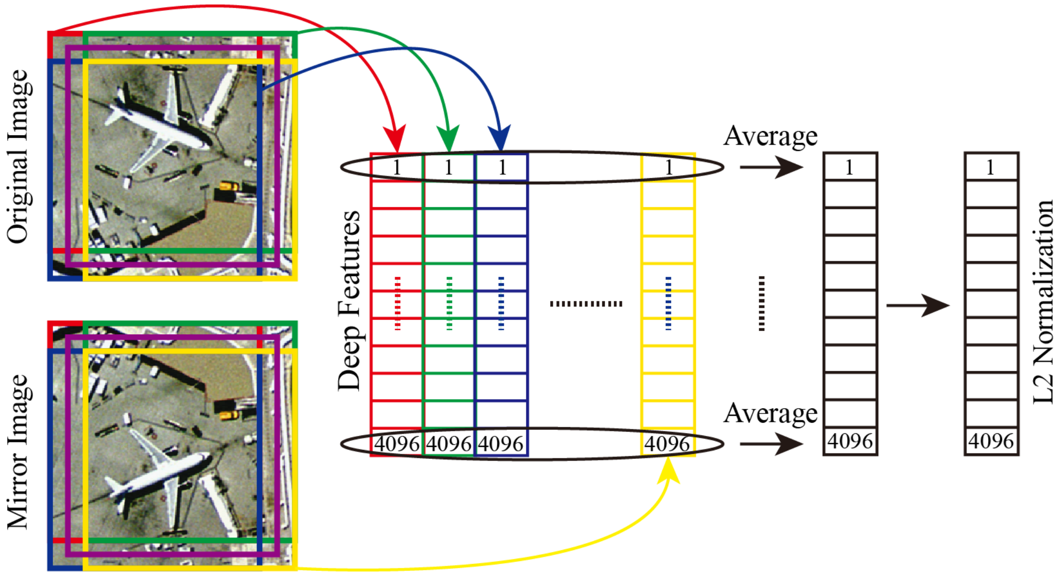


Figure 2. Data augmentation based on "center + corners with horizontal flips" strategy.

2.1.2. VGG-VD

VGG-VD won the localization and classification tracks in the ILSVRC 2014 competition. VGG-VD consists of VGG-VD16 and VGG-VD19. Giving the similar performance of both networks, VGG-VD16 is employed in this study.

VGG-VD16 consists of 13 convolutional layers, five pooling layers, and three fully connected layers. The input image size is 224 × 224 pixels with three channels (red-green-blue). The kernel size of each convolutional layer is 3 × 3 pixels, which is the smallest receptive field size. The convolution stride is set to 1 pixel, and the size of the feature maps is preserved after convolution extraction with spatial padding (1 pixel). Max pooling is used in five pooling layers, which follow some of the convolutional layers (not all the convolutional layers are followed by pooling). The size of a pooling region is set to 2 × 2 pixels with a stride of 2. For the three fully connected layers, the first two fully connected layers contain 4096 neurons; the third fully connected layer contains 1000-way (corresponding to 1000 neurons) ILSVRC classification. Similar to the feature extraction of CaffeNet, we extract two 4096-dimensional activation vectors from the first and second fully connected layers of VGG-VD16, followed by data augmentation and L2 normalization.

2.2. *Supervised DCF Learning*

Given a 4096-dimensional activation vector extracted from a pretrained CNN (CaffeNet or VGG-VD16), we can rearrange it into the form of an image matrix with the use of row priority or column priority, thereby resulting in a feature map with 64 × 64 pixels. The rearrangement of activation vector aims to learn DCF kernels with spatial arrangement characteristics.

To learn the DCF kernels, the 64 × 64-pixel feature map is first divided into a spatial arrangement of local patches through sliding window strategy. Given a set of local patches $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ extracted from the same spatial location with respect to a set of training feature maps (training images), each local patch $\mathbf{x}_i$ with $r \times r$ pixels belongs to a specific class of $\mathbf{C} = \{c_1, c_2, \cdots, c_K\}$. Through an unknown function f, these local patches can be mapped into other representations that satisfy the objective function to minimize in the L2-distance. The objective function can be defined as follows:

$$f_{opt} = \arg\min \frac{\sum_{c_k \in \mathbf{C}} \sum_{i \in c_k, j \in c_k} \left\| f(\mathbf{x}_i) - f(\mathbf{x}_j) \right\|^2}{\sum_{c_k \in \mathbf{C}} \sum_{m \in c_k, n \notin c_k} \left\| f(\mathbf{x}_m) - f(\mathbf{x}_n) \right\|^2}, \tag{1}$$

where the numerator measures the within-class distance and the denominator measures the between-class distance. Here, we seek a linear transformation (linear filter) that maps these patches to a new representation such that the L2-distance of the within-class is minimized while the L2-distance of the between-class is maximized. Thus, Equation (1) can be described as

$$\mathbf{K}_{opt} = \arg\min \frac{\sum_{c_k \in \mathbf{C}} \sum_{i \in c_k, j \in c_k} \left\| \mathbf{x}_i \otimes \mathbf{K} - \mathbf{x}_j \otimes \mathbf{K} \right\|^2}{\sum_{c_k \in \mathbf{C}} \sum_{m \in c_k, n \notin c_k} \left\| \mathbf{x}_m \otimes \mathbf{K} - \mathbf{x}_n \otimes \mathbf{K} \right\|^2}. \tag{2}$$

where $\otimes$ is the convolution operator, and $\mathbf{K}$ is the DCF kernel that we need to learn.

To learn $\mathbf{K}$ in Equation (2), we need to keep $\mathbf{K}$ as a vector form. Thus, we transform $\mathbf{x}_i$ into a new representation $\mathbf{A}_i$ (Figure 3), such that $\mathbf{x}_i \otimes \mathbf{K} = \mathbf{A}_i \overline{K}$, where $\overline{K}$ is the vectorized form of $\mathbf{K}$. For a local patch $\mathbf{x}_i$ with $r \times r$ pixels and a filter $\mathbf{K}$ with $w \times w$ pixels, the transformed matrix $\mathbf{A}_i$ with $r^2 \times w^2$ dimensions can be constructed by vectorizing the neighborhoods of $w \times w$ dimensions at each pixel in $\mathbf{x}_i$, as shown in Figure 3. Thus, we can obtain the following equation by substituting the convolution representation of Equation (2):

$$\overline{K}_{opt} = \frac{\sum_{c_k \in \mathbf{C}} \sum_{i \in c_k, j \in c_k} \left\| \mathbf{A}_i \overline{K} - \mathbf{A}_j \overline{K} \right\|^2}{\sum_{c_k \in \mathbf{C}} \sum_{m \in c_k, n \notin c_k} \left\| \mathbf{A}_m \overline{K} - \mathbf{A}_n \overline{K} \right\|^2}. \tag{3}$$

Equation (3) can be written as $\overline{K}_{opt} = \frac{\overline{K}^{\mathrm{T}} \mathbf{S_W} \overline{K}}{\overline{K}^{\mathrm{T}} \mathbf{S_B} \overline{K}}$, where $\mathbf{S_W} = \sum_{c_k \in \mathbf{C}} \sum_{i \in c_k, j \in c_k} (\mathbf{A}_i - \mathbf{A}_j)^{\mathrm{T}} (\mathbf{A}_i - \mathbf{A}_j)$ and $\mathbf{S_B} = \sum_{c_k \in \mathbf{C}} \sum_{m \in c_k, n \notin c_k} (\mathbf{A}_m - \mathbf{A}_n)^{\mathrm{T}} (\mathbf{A}_m - \mathbf{A}_n)$. $\mathbf{S_W}$ and $\mathbf{S_B}$ are symmetric matrices with w^2 dimensions. The minimum of Equation (3) can be obtained by solving the generalized eigenvalue problem. Thus, the minimum of $\overline{K}_{opt}$ is given by the minimum eigenvalue of $\mathbf{S_B}^{-1} \mathbf{S_W}$ and $\overline{K}$ equals the corresponding eigenvector (which can be reshaped to a matrix form).

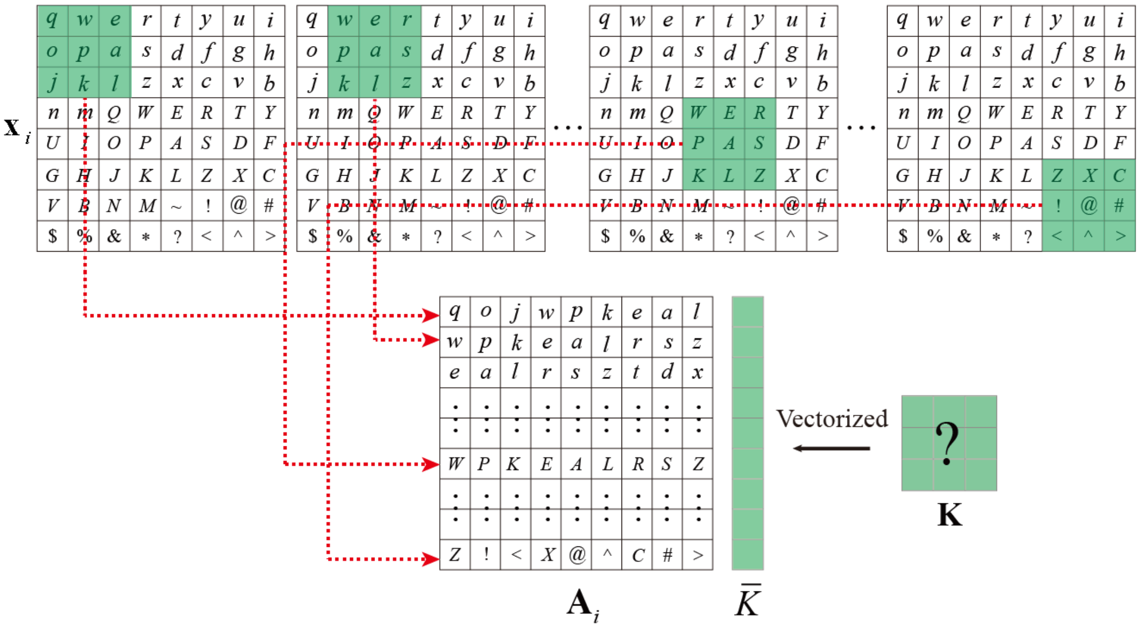


Figure 3. Transformed matrix $\mathbf{A}_i$ for a local patch with 8 × 8 pixels and a DCF kernel $\mathbf{K}$ with 3 × 3 pixels. In the first row, nine neighborhoods of the local patch are highlighted. The nine neighborhoods are concatenated to form a row of $\mathbf{A}_i$.

2.3. Transformation of Deep Features with DCF

Given a 64 × 64-pixel feature map (deep features) and a set of learned DCF kernels, the first step is to divide the feature map into equal-sized local patches ($r \times r$ pixels). We allow local patches to overlap with sliding stride of s pixels, resulting in a total number of $(\frac{64-r}{s}+1) \times (\frac{64-r}{s}+1)$ local patches. Correspondingly, $(\frac{64-r}{s}+1) \times (\frac{64-r}{s}+1)$ DCF kernels can be learned individually, according to Section 2.2.

To obtain the new representation, the $(\frac{64-r}{s}+1) \times (\frac{64-r}{s}+1)$ local patches are convolved with the corresponding DCF kernel to obtain the convolutional results, followed by feature concatenation, as shown in Figure 1. The convolution of each local patch is independent. During the convolution, the border pixels of each local patch are padded with zeros, thereby resulting in a D-dimensional new representation, where $D = (\frac{64-r}{s}+1) \times (\frac{64-r}{s}+1) \times r \times r$.

3. Experiments and Discussion

Experiments are conducted on two publicly available datasets, namely, 21-class land use dataset (denoted by 21-class) and 19-class satellite scene dataset (denoted by 19-class), to illustrate the effectiveness of the proposed DCF in improving the classification performances of deep features, as shown in Figure 4.

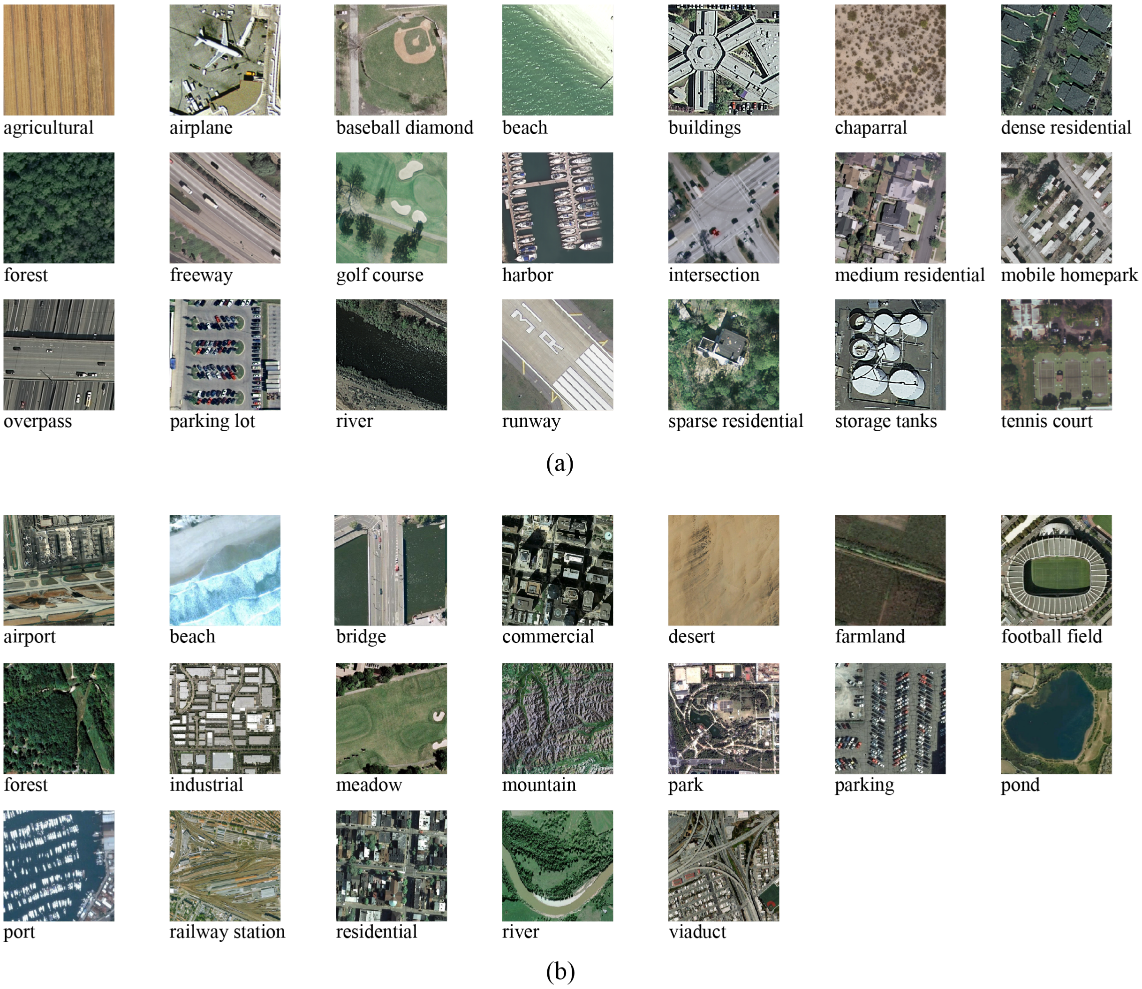


Figure 4. Datasets. (**a**) 21-class land use dataset; (**b**) 19-class satellite scene dataset.

3.1. Datasets and Experimental Configurations

3.1.1. Datasets

The 21-class dataset [3], which was downloaded from the United States Geological Survey National Map, was acquired from aerial orthoimagery with a pixel resolution of 1 foot. This dataset contains 2100 RGB (Red–Green–Blue) scene images (100 images per class) with a size of 256 × 256 pixels and covers multiple regions of the United States, as shown in Figure 4a. Various spatial patterns, homogeneous texture and color, and some land cover and possibly object classes exist in this dataset. The 21 classes are "agricultural", "airplane", "baseball diamond", "beach", "buildings", "chaparral", "dense residential", "forest", "freeway", "golf course", "harbor", "intersection", "medium density residential", "mobile home park", "overpass", "parking lot", "river", "runway", "sparse residential", "storage tanks", and "tennis courts". The 19-class dataset [34], which was acquired from Google Earth (Google Inc., Menlo Park, CA, America), consists of high-resolution satellite scene images up to half a meter. This dataset contains 950 RGB images (50 images per class) and covers a number of regions around the world, as shown in Figure 4b. The 19 classes are "airport", "beach", "bridge", "commercial", "desert", "farmland", "football field", "forest", "industrial", "meadow", "mountain", "park", "parking", "pond", "port", "railway station", "residential", "river", and "viaduct", respectively. In general, the images in both the 21-class and 19-class datasets are composed of various objects with changes in scales, rotations, orientations, and illuminations.

3.1.2. Experimental Configurations

Given a feature map with 64 × 64 pixels (reshaped by 4096-dimensional deep features), a 8 × 8-pixel local patch, which performs best in the case of 64 × 64-pixel feature map, is employed to extract the spatial arrangement local patches. DCF learning is then performed on each spatial location individually. After applying the learned DCF to a given feature map, the dimensionality of the final representation is determined by the sampling (sliding window) stride (s), which is analyzed in subsequent experiments.

For both 21-class and 19-class datasets, all results are repeated 10 times to report the average classification accuracy (denoted by *mean*) and standard deviation (denoted by *std*). In each round of testing, a fixed number of training images are randomly selected from each class and linear SVM [29] is employed for training, the overall accuracy of the remaining images (the so-called testing images) is used for evaluation. To obtain the overall accuracy, we count the number of correct classification images from a set of k testing images. The correct classification indicates that an image, which belongs to the c_i th class, is classified to the c_i th class through SVM prediction. Suppose that k' testing images are classified correctly. Then, the overall accuracy can be computed by $100 \times \frac{k'}{k}$. For the 10 rounds of testing, 10 overall accuracies (e.g., $a_1, a_2, ..., a_{10}$) can be obtained. The average classification accuracy *mean* and the standard deviation *std* can then be represented by $mean = \frac{1}{10} \sum_{m=1}^{10} a_m$ and $std = (\frac{1}{10-1} \sum_{m=1}^{10} (a_m - mean)^2)^{\frac{1}{2}}$, respectively.

3.2. DCF Kernel Size and Sampling Stride

Given the use of 8 × 8 pixel local patch, Figure 5 compares the effects of different filter sizes and sampling strides on the classification performances of both RSI datasets (under 10% training images per class). On the one hand, given a feature map with 64 × 64 pixels and a local patch with 8 × 8 pixels, the DCF kernel size with 3 × 3 pixels performs best on both datasets. Anything larger than the kernel size of 3 × 3 pixels overfits a local patch with 8 × 8 pixels. For example, the comparison between "filter size @ 3 × 3" and "filter size @ 5 × 5" indicates decreased classification accuracy with the increase in DCF kernel size.

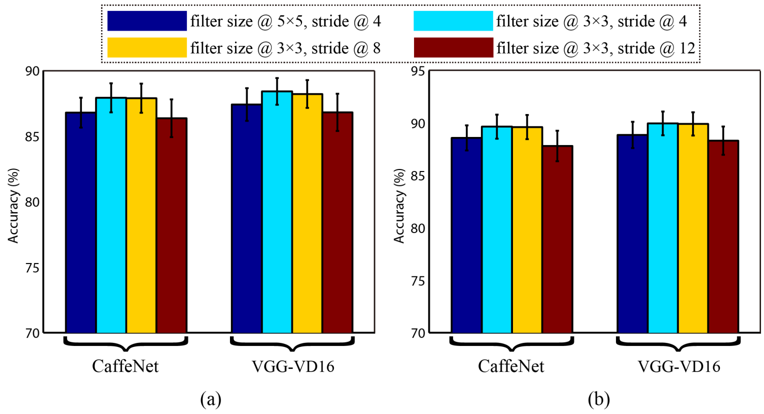


Figure 5. Effects of different DCF kernel sizes and sampling strides on classification performance. (**a**) 21-class dataset; (**b**) 19-class dataset.

On the other hand, the classification performances with respect to three types of sampling stride ("stride @ 4," "stride @ 8," and "stride @ 12") are compared. A large sampling stride corresponds to low dimensions for the final image representation. Given that the local patch size is 8 × 8 pixels, $s = 8$ indicates that the local patches are extracted by using non-overlapping strategy. Thus, a total of 4096-dimensional features can be obtained after DCF transformation, as the dimension length of the original deep features. Although the dimensionality of the final image representation reduces with the increase in s, the classification accuracy tends to decrease. In addition, the selection of $s = 4$ does not indicate accuracy advantages over $s = 8$ on both datasets. Compared with the selection of $s = 8$ (a total number of 64 filters), the selection of $s = 4$ would obviously increase the total number of DCF kernels that we need to learn, thereby increasing the burden of DCF training. In general, the selection of $s = 8$ is a good choice, considering the computational efficiency and classification performance.

With the use of $s = 8$, Figure 6 shows the differences between deep features (extracted by CaffeNet or VGG-VD16) with and without DCF. Given an input image, the 4096-dimensional deep features, which are non-negative, contain a large number of zero values (black pixels) because a ReLU operation is performed during CNN extraction. By contrast, DCF can produce a large number of non-zero values (positive and negative values) on the basis of the deep features because the DCF kernel summarizes adjacent deep features after rearrangement for each spatial location through linear convolution. In general, DCF extends the representation range of deep features from non-negative to real numbers.

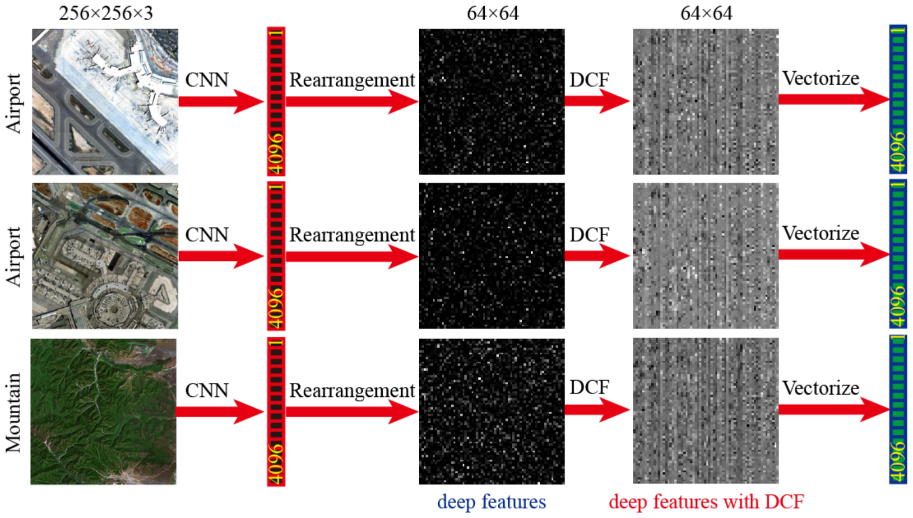
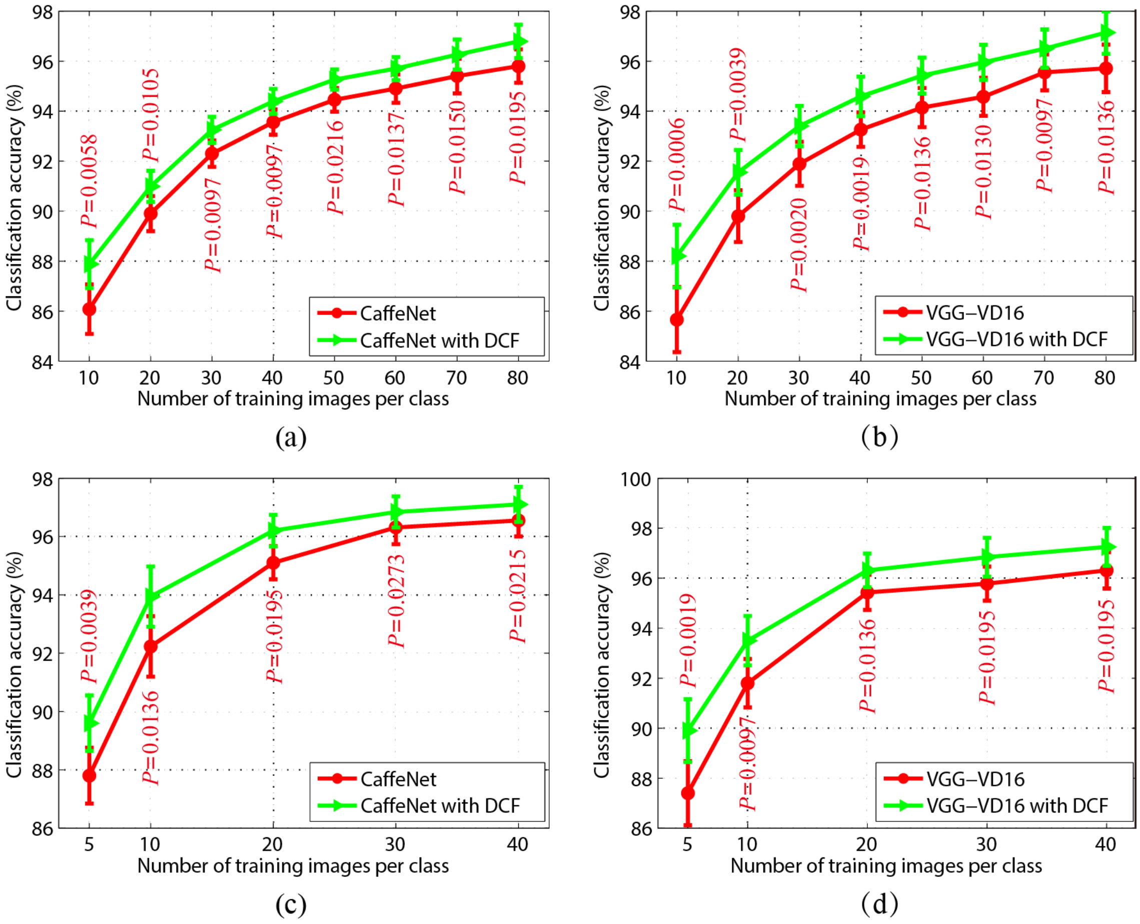


Figure 6. Differences between deep features with and without DCF.

3.3. Effectiveness of DCF for Deep Features

With the use of $s = 8$, Figure 7 shows the comparisons of the proposed method with and without DCF to illustrate the improvements of DCF for deep features. The "CaffeNet" or "VGG-VD16" shown in the legend of Figure 7 illustrates that the deep features (without DCF transformation) are directly used for SVM training and classification. In general, the following conclusions can be drawn: (1) DCF helps improve the classification performances of both pretrained CNNs on the two RSI datasets. (2) With the use of DCF, accuracy improvements can be obtained for different numbers of training images, especially for the case with fewer training images. (3) The selection of 10 training images per class in the 21-class dataset or the selection of five training images per class in the 19-class dataset indicates more improvements than other training ratios because the classification accuracies are close to saturation under a large number of training images (e.g., 80 training images per class in the 21-class dataset or 40 training images per class in the 19-class dataset). Overall, Figure 7 indicates that the patch-based DCF learning can substantially improve the separability of deep features without increasing dimensionality (e.g., $s = 8$).



(a) (b) (c) (d)

Figure 7. Effectiveness of DCF for deep features. *P* is the probability value obtained by Wilcoxon's signed-rank test. (**a**) CaffeNet-based deep features on the 21-class dataset; (**b**) VGG-VD16-based deep features on the 21-class dataset; (**c**) CaffeNet-based deep features on the 19-class dataset; (**d**) VGG-VD16-based deep features on the 19-class dataset.

With an increase in the number of training images per class, the accuracy improvement of DCF decreases. To illustrate that DCF can substantially improve classification performances under different numbers of training images, a statistical significance testing method called Wilcoxon's signed-rank test [35] is used. Given two methods (e.g., deep features with and without DCF), Wilcoxon's signed-rank test analyzes the paired classification accuracies for the 10 rounds of testing. If a substantial accuracy improvement is observed by DCF, then most results obtained by the deep features with DCF will be greater than those obtained by the deep features without DCF and those not greater will be smaller by only a small amount.

Wilcoxon's signed-rank test outputs a probability value P, which is the probability of observing an effect given that the null hypothesis [35] is true. Wilcoxon's signed-rank test analyzes whether the null hypothesis should be rejected. The observed result is statistically significant if the null hypothesis is rejected. Particularly, the null hypothesis can be rejected if P is less than a pre-defined significance level, which is usually set to 0.05 or 0.01 (significance level). As shown in Figure 7, all P values are smaller than 0.05, and these findings indicate that the difference between the deep features with and without DCF is significant despite some small accuracy improvements. In general, the deep features obtained by CaffeNet or VGG-VD16 with DCF are substantially better than those obtained without DCF.

With the use of 10% training images (10 images per class for 21-class and five images per class for 19-class), we further compare the classification performance of each class between deep features with and without DCF transformation, as shown in Figure 8. Overall, DCF helps improve the classification performance of those categories that contain buildings or significant features (e.g., objects or textures). For the 21-class dataset, CaffeNet-based deep features with DCF indicates obvious accuracy advantages on "golf course," "medium density residential," "overpass," "river," and "storage tanks" compared with that without DCF. VGG-VD16-based deep features with DCF indicates obvious accuracy advantages on "density residential," "intersection," "river," "runway," and "storage tanks" compared with that without DCF. For the 19-class dataset, CaffeNet-based deep features with DCF indicates obvious accuracy advantages on "commercial," "forest," and "industrial" compared with that without DCF, and VGG-VD16 with DCF indicates obvious accuracy advantages on "airport," "industrial," "mountain," "pond," and "residential" compared with that without DCF.

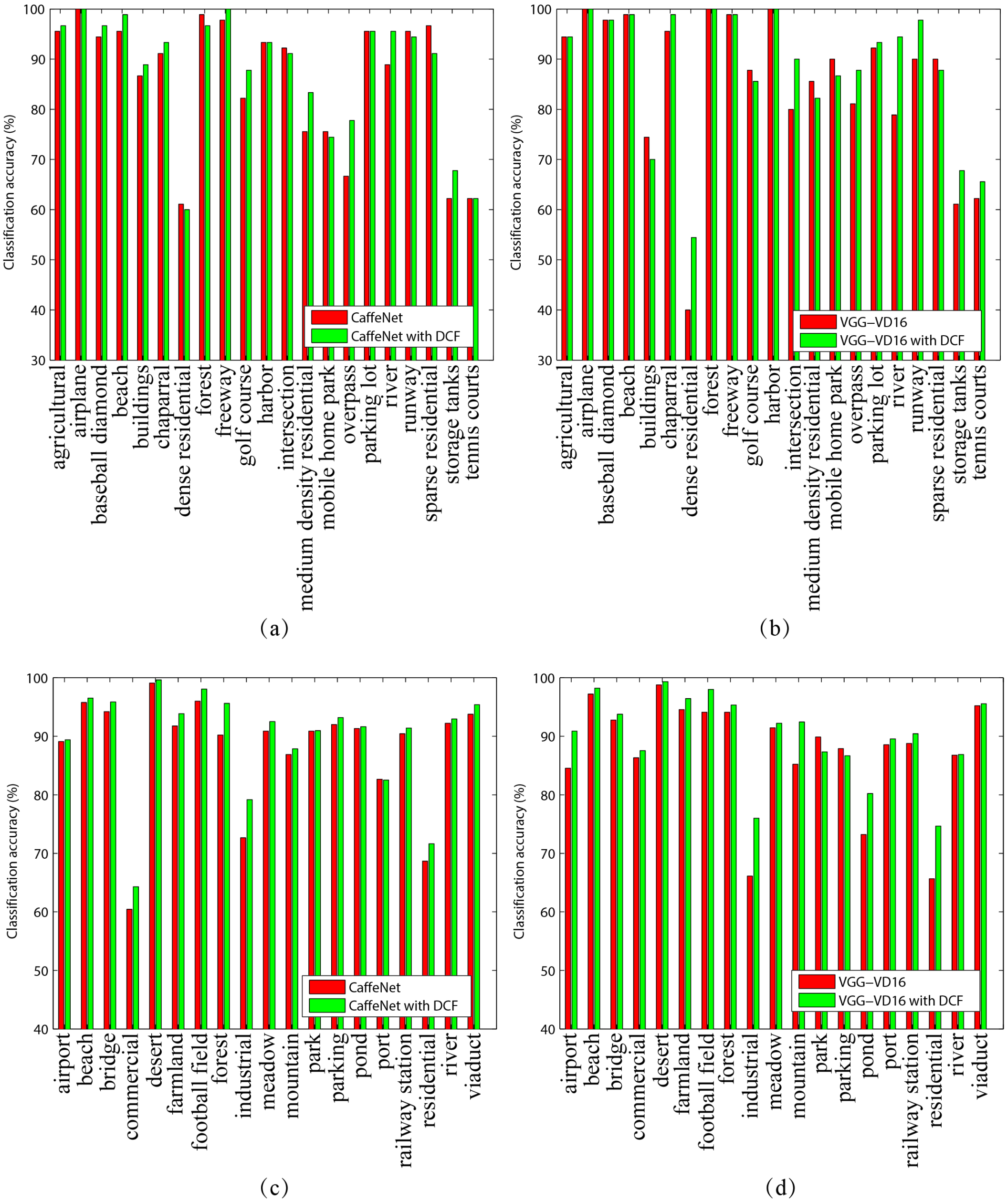
3.4. Analysis of Confusion Matrix

Figure 8 compares the classification performance of each class between deep features with and without DCF and identifies which classes can be improved by DCF transformation. However, the factors that affect the classification performance of each class are not clear. In this section, four confusion matrices (CaffeNet with DCF and VGG-VD16 with DCF on two datasets) are provided to determine the classes that easily produce classification confusion, thereby helping explain the effects of other classes on a given class, as shown in Figures 9 and 10. Unlike the results in Figure 8, which are based on selecting 10% training images per class and testing the remaining 90% images on each dataset, the four confusion matrices in Figures 9 and 10 are based on selecting 80% training images per class and testing the remaining 20% images.

Figure 9 shows two confusion matrices for the 21-class dataset. For Figure 9a,b, "agricultural," "airplane," "beach," "chaparral," "forest," "golf course," "harbor," "parking lot," and "river" achieve high classification accuracies. The well-performing classes have different characteristics. For example, images in "agricultural" and "forest" have significant textures; images in "airplane" have significant aircrafts that are easy to distinguish from other objects (e.g., buildings); images in "beach" show significant color features; and images in "harbor" or "parking lot" have significant spatial and texture structures. In contrast, some classes perform poorly, such as "dense residential," "medium density residual," "sparse residential," and "tennis courts." These classes have similar characteristics, such as the presence of various buildings that often have similarities across different classes. "Tennis courts" performs poorly because the tennis courts are surrounded by buildings and are generally unremarkable.

In general, the overall classification performances of both CNNs are similar, but the accuracy performances on several classes (e.g., "storage tanks") are different between CaffeNet and VGG-VD16. Although VGG-VD16 is much deeper than CaffeNet in the network architecture, the overall performance of the former does not indicate advantages over the latter due to the accuracy saturation of this dataset.

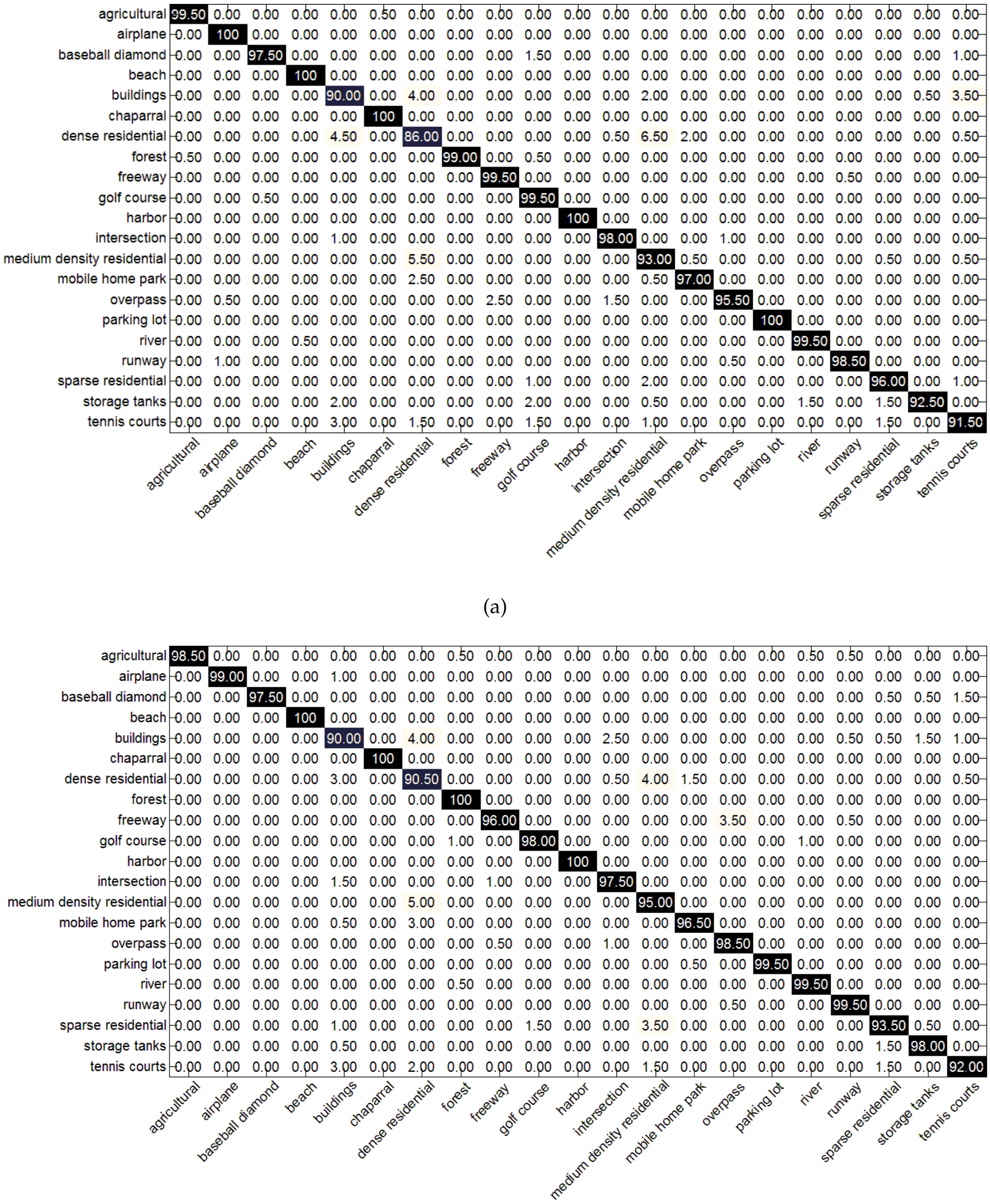
Similar to Figures 9 and 10 shows two confusion matrices of the 19-class dataset. For Figure 10a,b, "airport," "beach," "desert," "football field," "meadow," "pond," "river," and "viaduct" indicate high classification performances. Similar to the 21-class dataset, the well-performing classes in the 19-class dataset contain significant textures, colors, significant objects, or spatial structures. "Commercial" performs poorly in Figure 10a,b because numerous buildings exist in this class. In general, the two types of pretrained CNN show similar classification performances to that in Figure 9.



(a) (b)

(c) (d)

Figure 8. Comparisons of the classification performance for each class between deep features with and without DCF using 10% training images per class. (**a**) CaffeNet-based deep features on the 21-class dataset; (**b**) VGG-VD16-based deep features on the 21-class dataset; (**c**) CaffeNet-based deep features on the 19-class dataset; (**d**) VGG-VD16-based deep features on the 19-class dataset.



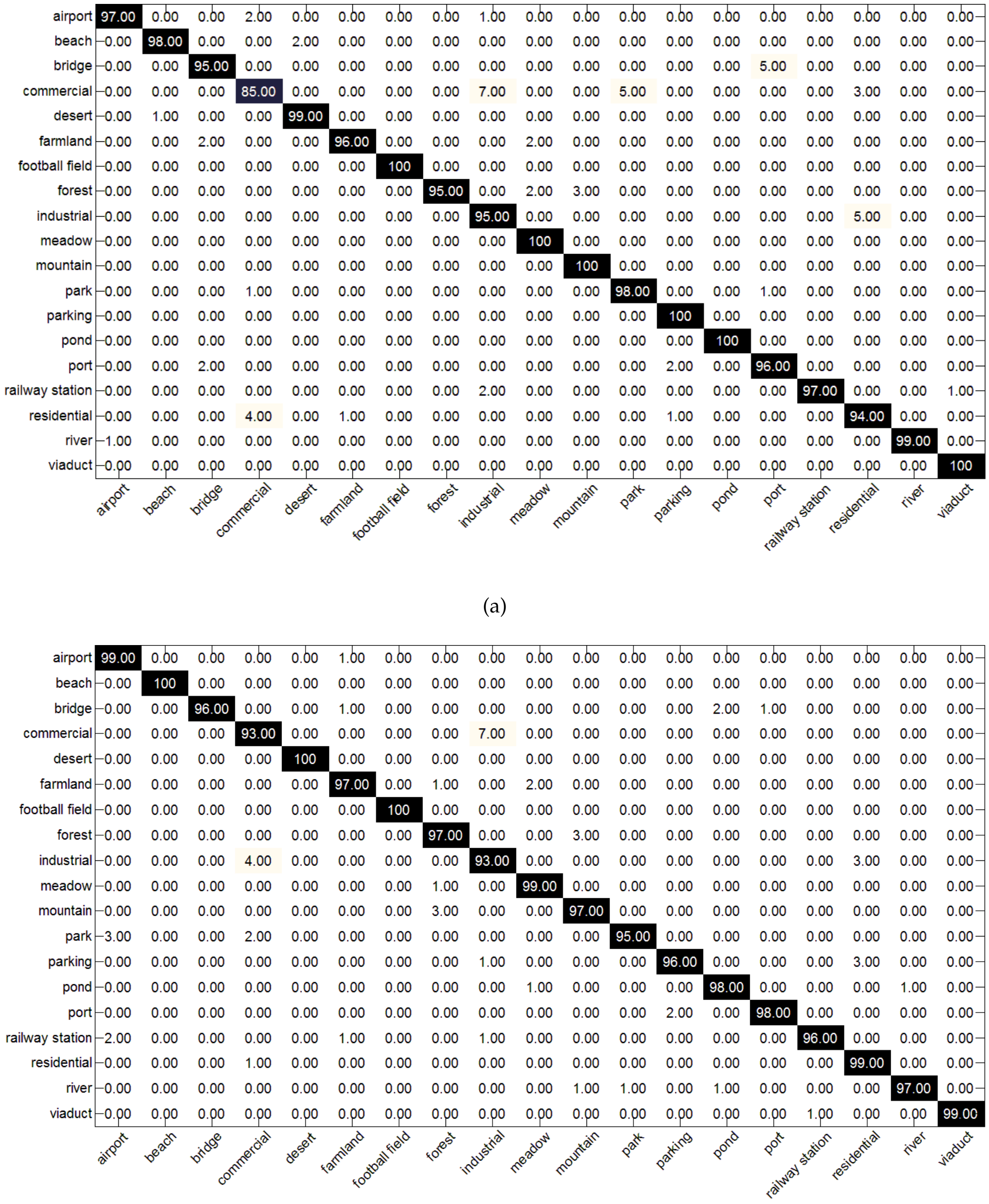
	agricultural	airplane	baseball diamond	beach	buildings	chaparral	dense residential	forest	freeway	golf course	harbor	intersection	medium density residential	mobile home park	overpass	parking lot	river	runway	sparse residential	storage tanks	tennis courts
agricultural	99.50	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
airplane	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
baseball diamond	0.00	0.00	97.50	0.00	0.00	0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
beach	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
buildings	0.00	0.00	0.00	0.00	90.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	3.50
chaparral	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dense residential	0.00	0.00	0.00	0.00	4.50	0.00	86.00	0.00	0.00	0.00	0.00	0.50	6.50	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
forest	0.50	0.00	0.00	0.00	0.00	0.00	0.00	99.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
freeway	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00
golf course	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	99.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
harbor	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
intersection	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	98.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
medium density residential	0.00	0.00	0.00	0.00	0.00	0.00	5.50	0.00	0.00	0.00	0.00	0.00	93.00	0.50	0.00	0.00	0.00	0.00	0.50	0.00	0.50
mobile home park	0.00	0.00	0.00	0.00	0.00	0.00	2.50	0.00	0.00	0.00	0.00	0.00	0.50	97.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
overpass	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	2.50	0.00	0.00	1.50	0.00	0.00	95.50	0.00	0.00	0.00	0.00	0.00	0.00
parking lot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00
river	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.50	0.00	0.00	0.00	0.00
runway	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	98.50	0.00	0.00	0.00
sparse residential	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	96.00	0.00	1.00
storage tanks	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.50	0.00	0.00	0.00	1.50	0.00	1.50	92.50	0.00
tennis courts	0.00	0.00	0.00	0.00	3.00	0.00	1.50	0.00	0.00	1.50	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.50	0.00	91.50

(a)

	agricultural	airplane	baseball diamond	beach	buildings	chaparral	dense residential	forest	freeway	golf course	harbor	intersection	medium density residential	mobile home park	overpass	parking lot	river	runway	sparse residential	storage tanks	tennis courts
agricultural	98.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00
airplane	0.00	99.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
baseball diamond	0.00	0.00	97.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	1.50
beach	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
buildings	0.00	0.00	0.00	0.00	90.00	0.00	4.00	0.00	0.00	0.00	0.00	2.50	0.00	0.00	0.00	0.00	0.00	0.50	0.50	1.50	1.00
chaparral	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dense residential	0.00	0.00	0.00	0.00	3.00	0.00	90.50	0.00	0.00	0.00	0.00	0.50	4.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50
forest	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
freeway	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	96.00	0.00	0.00	0.00	0.00	0.00	3.50	0.00	0.00	0.50	0.00	0.00	0.00
golf course	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	98.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
harbor	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
intersection	0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	1.00	0.00	0.00	97.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
medium density residential	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	95.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mobile home park	0.00	0.00	0.00	0.00	0.50	0.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	96.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
overpass	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	1.00	0.00	0.00	98.50	0.00	0.00	0.00	0.00	0.00	0.00
parking lot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	99.50	0.00	0.00	0.00	0.00	0.00
river	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.50	0.00	0.00	0.00	0.00
runway	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	99.50	0.00	0.00	0.00
sparse residential	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1.50	0.00	0.00	3.50	0.00	0.00	0.00	0.00	0.00	93.50	0.50	0.00
storage tanks	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.50	98.00	0.00
tennis courts	0.00	0.00	0.00	0.00	3.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	1.50	0.00	92.00

(b)

Figure 9. Confusion matrices for the 21-class dataset in the case of 80 training images per class. All results are given as percentages, and the rows and columns represent the ground truth and classification accuracies, respectively. (**a**) CaffeNet-based deep features with DCF; (**b**) VGG-VD16-based deep features with DCF.



	airport	beach	bridge	commercial	desert	farmland	football field	forest	industrial	meadow	mountain	park	parking	pond	port	railway station	residential	river	viaduct
airport	97.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
beach	0.00	98.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bridge	0.00	0.00	95.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00
commercial	0.00	0.00	0.00	85.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00
desert	0.00	1.00	0.00	0.00	99.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
farmland	0.00	0.00	2.00	0.00	0.00	96.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
football field	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	0.00	0.00	95.00	0.00	2.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
industrial	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	95.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00
meadow	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
park	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
parking	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00
pond	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00
port	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	96.00	0.00	0.00	0.00	0.00
railway station	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	97.00	0.00	0.00	1.00
residential	0.00	0.00	0.00	4.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	94.00	0.00	0.00
river	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.00	0.00
viaduct	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100

(a)

	airport	beach	bridge	commercial	desert	farmland	football field	forest	industrial	meadow	mountain	park	parking	pond	port	railway station	residential	river	viaduct
airport	99.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
beach	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bridge	0.00	0.00	96.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	1.00	0.00	0.00	0.00	0.00
commercial	0.00	0.00	0.00	93.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
desert	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
farmland	0.00	0.00	0.00	0.00	0.00	97.00	0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
football field	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	0.00	0.00	97.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
industrial	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	93.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00
meadow	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	99.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	97.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
park	3.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	95.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
parking	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	96.00	0.00	0.00	0.00	3.00	0.00	0.00
pond	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	98.00	0.00	0.00	0.00	1.00	0.00
port	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	98.00	0.00	0.00	0.00	0.00
railway station	2.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	96.00	0.00	0.00	0.00
residential	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.00	0.00	0.00
river	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	1.00	0.00	0.00	0.00	97.00	0.00
viaduct	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	99.00

(b)

Figure 10. Confusion matrices for the 19-class dataset in the case of 40 training images per class. (**a**) CaffeNet-based deep features with DCF; (**b**) VGG-VD16-based deep features with DCF.

3.5. Comparisons with Other Methods

Table 1 summarizes the performance comparisons of different methods using different training ratios. Among these comparison methods, EGTD [8] and multiple kernel learning (MKL) [6] are based on handcrafted features. EGTD computes the means and standard deviations of Gabor coefficients and the cross-correlation between these coefficients at different scales or orientations. MKL can determine a suitable combination of a set of handcrafted features automatically. fDNF + FV (fusion Divisive Normalization Features with Fisher Vector) [36] is a mid-level representation method based on local description and Fisher encoding, and the local descriptions are obtained by handcrafted features. Unsupervised feature learning (UFL) [12] consists of low-level feature extraction, feature learning, encoding, and pooling. UFL obtains sparse feature representations through encoding the low-level features with a set of learned basis functions, which are generated by unsupervised learning. GBRCN [21], LPCNN (Large Patch Convolutional Neural Networks) [37], CaffeNet [2], and VGG-VD16 [2] are four types of CNN-based methods. LPCNN investigated an appropriate model to balance the trade-off of CNN and limited trainable images. GBRCN can effectively combine numerous deep neural networks. CaffeNet and VGG-VD16 are based on the pretrained models (trained on ImageNet) to extract 4096-dimensional deep features from images, followed by SVM training and classification [2,22,24]. In Table 1, the results for EGTD, MKL, fDNF + FV, UFL, GBRCN, and LPCNN are obtained from the original references; the empty results (denoted by "–") indicate that the corresponding reference does not provide the results. CaffeNet, VGG-VD16, Proposed (CaffeNet with DCF), and Proposed (VGG-VD16 with DCF) are implemented using MatConvNet [32] . All methods are based on learning the training images with a fixed number of images per class and testing the overall classification performance of the remaining images.

Table 1. Comparisons of classification accuracy (%) with other methods.

		21-Class			**19-Class**		
Training Images Per Class	**10**	**50**	**80**	**5**	**20**	**25**	**40**
EGTD	–	87.56 ± 1.28	–	–	–	–	
MKL	74.57 ± 1.45	88.86 ± 0.90	91.84 ± 1.29	84.05 ± 1.67	–	–	–
fDNF + FV	–	91.09	93.62	–	–	94.69	–
UFL	–	–	81.67 ± 1.23	–	–	–	–
GBRCN	–	–	94.53	–	–	–	
LPCNN	–	–	89.90	–	–	–	–
CaffeNet	86.08 ± 0.98	94.45 ± 0.57	95.83 ± 0.69	87.84 ± 0.95	95.11 ± 0.59	–	96.55 ± 0.53
VGG-VD16	85.66 ± 1.15	94.14 ± 0.79	95.71 ± 0.89	87.43 ± 1.21	95.43 ± 0.71	–	96.31 ± 0.70
CaffeNet with DCF	87.88 ± 0.95	95.26 ± 0.50	96.79 ± 0.66	89.60 ± 0.93	96.20 ± 0.56	–	97.10 ± 0.51
VGG-VD16 with DCF	88.20 ± 0.99	95.42 ± 0.71	97.10 ± 0.85	89.90 ± 1.12	96.31 ± 0.67	–	97.31 ± 0.68

Several conclusions can be drawn from Table 1. First, local feature representation followed by feature encoding performs well among these methods based on handcrafted features. EGTD cannot encode local information because of the averaging of the wavelet coefficients on image domain. Second, although ULF can learn features from images automatically, it is an unsupervised learning method that cannot learn class-specific separable features unlike supervised CNNs. Feature encoding based on low-level features or UFL can only generate shallow-based mid-level features with limited representative ability, which essentially prevents them from achieving desirable performances. Third, CNN can obtain different levels of abstraction from the input image, ranging from low-level features in the initial layers, mid-level features in the intermediate layers, to high-level features in the final layers. To obtain an effective CNN model, the training samples play an important role in the CNN-based methods. Among these comparison methods, GBRCN and LPCNN are trained on the 21-class dataset, which contains only 2100 images. By contrast, CaffeNet and VGG-VD16 are trained on the ImageNet dataset with millions of images. Compared with GBRCN and LPCNN, the generality of both CaffeNet and VGG-VD16 is obvious. Finally, the comparison of CaffeNet and "Proposed (CaffeNet-DCF)," and the comparison of VGG-VD16 and "Proposed (VGG-VD16-DCF)" indicate that DCF can improve the classification accuracy on both datasets, especially with fewer training images.

4. Conclusions

In this study, we propose a novel method for RSI representation based on deep features and DCF kernels to improve the separability of the deep features extracted from CNN models. Given a pretrained CNN model, the deep features are represented by activation vectors extracted from fully connected layers. Then, the deep features are rearranged into the form of an image matrix to obtain a spatial arrangement of local patches. Finally, supervised DCF learning, which helps enhance the distinguishability of activation vectors, is performed on each spatial location individually to learn the corresponding DCF kernel. Experiments on two RSI datasets illustrate the effectiveness of the DCF in improving classification accuracies. In future works, we intend to investigate DCF learning for multiple CNNs simultaneously.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (No. 41571402), the Science Fund for Creative Research Groups of the National Natural Science Foundation of China (No. 61221003), and Shanghai Jiao Tong University Agri-X Fund (No. Agri-X2015004).

Author Contributions: Na Liu wrote the paper. Na Liu, Xiankai Lu, and Lihong Wan analyzed the data. Hong Huo and Tao Fang conceived of and designed the experiments. Na Liu and Lihong Wan performed the experiments. Lihong Wan contributed analysis tools.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
RSI	remote sensing image
DCF	discriminative convolution filter
SIFT	scale-invariant feature transform
HOG	histograms of oriented gradients
LBP	local binary patterns
BOW	bag-of-visual-words
EGTD	enhanced Gabor texture descriptor
GBRCN	gradient-boosting random convolutional network
SVM	support vector machine
ILSVRC	imageNet large scale visual recognition challenge
MKL	multiple kernel learning
UFL	unsupervised feature learning

References

1. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883.
2. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L. AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981.
3. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the Sigspatial International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
4. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
5. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *Isprs J. Photogr. Remote Sens.* **2014**, *98*, 119–132.
6. Cusano, C.; Napoletano, P.; Schettini, R. Remote Sensing Image Classification Exploiting Multiple Kernel Learning. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2331–2335.
7. Ren, J.; Jiang, X.; Yuan, J. Learning LBP structure by maximizing the conditional mutual information. *Pattern Recognit.* **2015**, *48*, 3180–3190.
8. Risojevic, V.; Babic, Z. Fusion of Global and Local Descriptors for Remote Sensing Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 836–840.
9. Risojevi, V.; Momić, S.; Babić, Z. Gabor descriptors for aerial image classification. In *International Conference on Adaptive and Natural Computing Algorithms*; Springer: Berlin, Germany, 2011; pp. 51–60.
10. Oliva, A.; Torralba, A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175.

11. Bao, C.; Ji, H.; Quan, Y.; Shen, Z. Dictionary Learning for Sparse Coding: Algorithms and Convergence Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1356–1369.
12. Cheriyadat, A.M. Unsupervised Feature Learning for Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 439–451.
13. Zhou, W.; Shao, Z.; Diao, C.; Cheng, Q. High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder. *Remote Sens. Lett.* **2015**, *6*, 775–783.
14. Othman, E.; Bazi, Y.; Alajlan, N.; Alhichri, H.; Melgani, F. Using convolutional features and a sparse autoencoder for land-use scene classification. *Int. J. Remote Sens.* **2016**, *37*, 2149–2167.
15. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
17. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778
20. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
21. Zhang, F.; Du, B.; Zhang, L. Scene Classification via a Gradient Boosting Random Convolutional Network Framework. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1793–1802.
22. Nogueira, K.; Penatti, O.A.B.; Santos, J.A.D. Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. *Pattern Recognit.* **2017**, *61*, 539–556.
23. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2014; pp. 818–833.
24. Penatti, O.A.B.; Nogueira, K.; Santos, J.A.D. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 44–51.
25. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
26. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu , M.; Fergus, R.; LeCun, Y. OverFeat: integrated recognition, localization and detection using convolutional networks. *arXiv* **2014**, arXiv:1312.6229.
27. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *Acta Ecol. Sin.* **2015**, *28*, 627–635.
28. Wan, L.; Liu, N.; Huo, H.; Fang, T. Selective convolutional neural networks and cascade classifiers for remote sensing image classification. *Remote Sens. Lett.* **2017**, *8*, 917–926.
29. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27.
30. Kumar, R.; Banerjee, A.; Vemuri, B.C.; Pfister, H. Trainable Convolution Filters and Their Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1423–1436.
31. Kumar, R.; Banerjee, A.; Vemuri, B.C. Volterrafaces: Discriminant analysis using Volterra kernels. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2010; pp. 150–155.
32. Vedaldi, A.; Lenc, K. MatConvNet: Convolutional Neural Networks for MATLAB. In Proceedings of the ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 689–692.
33. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv* **2014**, arXiv:1405.3531.

34. Sheng, G.; Yang, W.; Xu, T.; Sun, H. High-resolution satellite scene classification using a sparse coding based multiple feature combination. *Int. J. Remote Sens.* **2012**, *33*, 2395–2412.
35. Japkowicz, N.; Shah, M. *Evaluating Learning Algorithms: A Classification Perspective*; Cambridge University Press: Cambridge, UK, 2011.
36. Wan, L.; Liu, N.; Guo, Y.; Huo, H.; Fang, T. Local feature representation based on linear filtering with feature pooling and divisive normalization for remote sensing image classification. *J. Appl. Remote Sens.* **2017**, *11*, 016017.
37. Zhong, Y.; Fei, F.; Zhang, L. Large patch convolutional neural networks for the scene classification of high spatial resolution imagery. *J. Appl. Remote Sens.* **2016**, *10*, 025006.