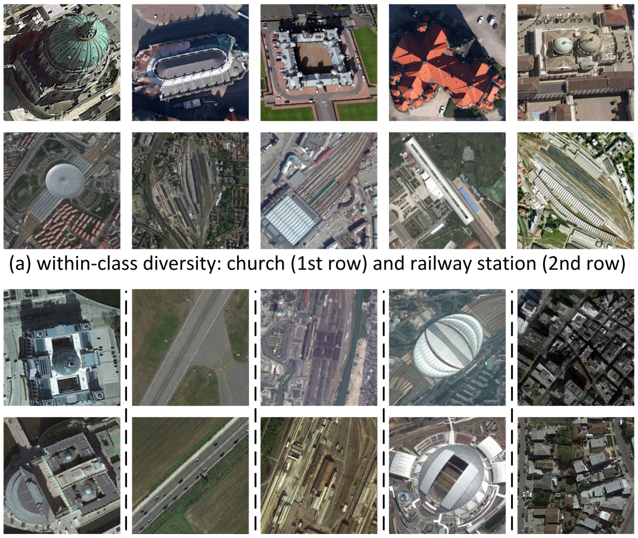
# When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs

Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han

***Abstract*—Remote sensing image scene classification is an active and challenging task driven by many applications. More recently, with the advances of deep learning models especially convolutional neural networks (CNNs), the performance of remote sensing image scene classification has been significantly improved due to the powerful feature representations learnt through CNNs. Although great success has been obtained so far, the problems of within-class diversity and between-class similarity are still two big challenges. To address these problems, in this paper, we propose a simple but effective method to learn discriminative CNNs (D-CNNs) to boost the performance of remote sensing image scene classification. Different from the traditional CNN models that minimize only the cross entropy loss, our proposed D-CNN models are trained by optimizing a new discriminative objective function. To this end, apart from minimizing the classification error, we also explicitly impose a metric learning regularization term on the CNN features. The metric learning regularization enforces the D-CNN models to be more discriminative so that, in the new D-CNN feature spaces, the images from the same scene class are mapped closely to each other and the images of different classes are mapped as farther apart as possible. In the experiments, we comprehensively evaluate the proposed method on three publicly available benchmark data sets using three off-the-shelf CNN models. Experimental results demonstrate that our proposed D-CNN methods outperform the existing baseline methods and achieve state-of-the-art results on all three data sets.**

***Index Terms*—Convolutional neural networks (CNNs), deep learning, discriminative CNNs (D-CNNs), metric learning, remote sensing image scene classification.**



(a) within-class diversity: church (1st row) and railway station (2nd row)

(b) between-class similarity: church *vs.* palace, runway *vs.* freeway, railway station *vs.* industrial area, railway station *vs.* stadium, commercial area *vs.* dense residential (from top to bottom and from left to right)

Fig. 1. Motivation of the proposed method. (a) Within-class diversity and (b) between-class similarity are two major challenges that often degenerate the scene classification performance. This motivates us to learn more D-CNN feature representations that have small within-class scatter but big between-class separation. These examples are from the challenging NWPU-RESISC45 data set [1].

## I. Introduction

REMOTE sensing image scene classification [1], [2], which focuses on classifying remote sensing images into a set of semantic classes according to the image contents, has been attracting more and more research interest driven by its broad applications such as object detection [3]–[5].

In the past decades, extensive efforts have been made in the development of various feature representations for the task of remote sensing image scene classification and a detailed review can be found in [1] and [2]. Especially in recent years, from the milestone work of convolutional neural networks (CNNs) for image classification on ImageNet large-scale visual recognition challenge [6], a variety of CNN-based methods [7]–[13] have been dominating the field of remote sensing image scene classification. The huge success of various CNN-based scene classification methods is largely due to the use of deep CNN models (e.g., AlexNet [6], VGGNet [14], and GoogLeNet [15]) to extract powerful image feature representations, which are more discriminative than hand-crafted low-level features such as color, texture, and spectral features.

However, although the performance of scene classification has been significantly improved due to the powerful features learnt through deep CNN models, the problems of within-class diversity and between-class similarity still remain two major challenges, as shown in Fig. 1. These challenges will generally degenerate the performance of remote sensing image scene classification. In this situation, how to learn more discriminative CNN (D-CNN) feature representations that have small within-class scatter but big between-class separation is highly appealing.

Manuscript received September 7, 2017; revised December 2, 2017; accepted December 10, 2017. Date of publication January 9, 2018; date of current version April 20, 2018. This work was supported in part by the National Science Foundation of China under Grant 61772425, Grant 61473231, Grant 61773315 and Grant 61701415, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2017JM6044, and in part by the Science and Technology Foundation. *(Corresponding author: Junwei Han.)*

The authors are with the School of Automation, Northwestern Polytechnical University, Xi’an 710072, China (e-mail: junweihan2010@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2783902

To explore a possible solution to deal with the challenges, this paper proposes an effective method to learn D-CNNs to improve the performance of remote sensing image scene classification. This is achieved by developing a new objective function based on the existing CNN models but without changing their architectures. To this end, apart from minimizing the cross entropy loss, we explicitly impose a metric learning regularization term on the CNN features to enforce the D-CNN models to be more discriminative. Thus, in the D-CNN feature spaces, the images from the same scene class are as close as possible and the images of different classes are as far away as possible. In the experiments, we implement our proposed algorithm using three off-the-shelf CNN models including AlexNet [6], VGGNet [14], and GoogLeNet [15] and evaluate their performances on three publicly available benchmark data sets. Experimental results demonstrate that our proposed D-CNN features outperform the existing baseline methods and achieve state-of-the-art results on all three data sets.

The rest of this paper is organized as follows. Section II gives a brief review of related works. Section III briefly introduces the preliminary knowledge including CNNs and metric learning. Section IV describes the proposed method in detail. Section V presents comprehensive experimental results on three publicly available data sets. Finally, Section VI concludes this paper.

## II. Related Work

Scene classification using remote sensing images has been extensively studied for the last few decades owing to its broad applications. Based on the features used for scene classification, the existing approaches can be roughly categorized into three main categories [1]: handcrafted feature-based methods [16]–[24], unsupervised feature learning-based methods [25]–[36], and deep feature learning-based methods [7]–[12], [37]–[43].

Handcrafted feature-based methods [16]–[22] mainly focus on designing various human-engineering features, such as spectral, color, texture, and shape information or their combination, to represent the primary characteristic of a scene image. Most of the early remote sensing image scene classification methods fall within this category. Among all kinds of handcrafted features, color feature, texture feature [16]–[18], [23], and histogram of oriented gradients-based shape descriptors [19]–[22] are the most widely used feature representations. However, in practical applications, the performance is largely limited by the hand-crafted descriptors as they are difficult to describe the rich semantic information contained in remote sensing images.

To fix the limitation of handcrafted feature based methods, many research works have developed unsupervised feature learning based methods for scene classification [25]–[35]. Rather than devoting to the design of handcrafted feature descriptors, unsupervised feature learning-based methods mainly aim at learning a set of basis functions used for feature encoding. For unsupervised feature learning methods, the input is a set of handcrafted feature descriptors such as scale-invariant feature transform and the output is a set of learned features. By learning features from images to replace handcrafted features, one can expect obtaining more discriminative features that are better suited for the representation of the scene images. Typical unsupervised feature learning methods include, but not limited to, principal component analysis, k-means clustering, sparse coding [26]–[28], [33], [44] and autoencoder [29], [31]. In addition, it is worth noting that the widely used codebook in bag of visual words-based methods [45]–[50] is often generated by performing unsupervised k-means clustering on the set of handcrafted feature descriptors or raw pixel intensity values. When compared with handcrafted feature-based methods, unsupervised feature learning methods have obtained good performance. However, most of the unsupervised feature learning methods do not make use of the scene class information and so cannot guarantee the best discrimination ability between different scene classes.

In more recent years, thanks to the availability of large-scale training data and the advance of high-performance computing units [51], [52], deep feature learning-based methods [7]–[12], [37]–[42] have been attracting more research attention. This kind of method automatically learns features from the raw input data using deep-architecture neural networks such as stacked autoencoder [37] and CNNs [7]–[12], [38]–[42]. Deep feature learning-based methods have two important advantages against the above-mentioned two kinds of methods. On the one hand, compared with handcrafted feature-based methods that generally involve abundant engineering skills and domain expertise [16]–[22], deep learning features are directly generated from raw data via neural networks of deep architecture. Thus, the burden for human engineering-based feature design has been transferred to the network construction. On the other hand, in comparison with unsupervised feature learning-based methods [25]–[35] that heavily rely on shallow-structured models (e.g., sparse coding) [25]–[34], deep features are more powerful via the use of multiple stacked feature extraction layers.

Although deep learning especially CNN feature-based methods [7]–[12], [38]–[42] have achieved huge success for scene classification, most of them regard only CNN models as feature extractors and few concerns are concentrated on the design of the object functions of CNN models. Different from previous works, this paper mainly focuses on enriching the discrimination of the CNN feature representations by optimizing a new object function. Our proposed method is implemented based on three widely used CNN models including AlexNet [6], VGGNet [14], and GoogLeNet [15], without changing the model architectures, which are most related with our work and therefore are also the baseline methods in our experiments.

## III. Preliminary Knowledge

This paper mainly focuses on training D-CNN models by embedding a metric learning regularization term on the object function of the traditional CNN models. Therefore, this section will briefly introduce some preliminary knowledge including CNNs and metric learning.

### A. Convolutional Neural Networks

Recently, CNN has shown impressive performance in many applications including remote sensing image

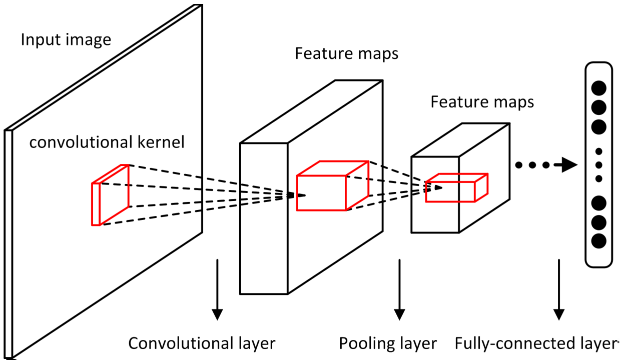


Fig. 2. Typical architecture of a CNN, which is structured as a series of layers including convolutional layers, pooling layers, and FC layers.

analysis [7], [10], [13], [42], [53]–[59]. As shown in Fig. 2, the typical architecture of a CNN is structured as a series of layers including convolutional layers, pooling layers, and fully connected (FC) layers [6], [14], [15].

*1) Convolutional Layers:* They are the most important layers for feature extraction. The first convolutional layers generally capture low-level features and the deeper layers are capable of extracting high-level features by combining low-level ones. Each unit in a convolutional layer is connected to a local patch in the feature maps of the previous layer through a set of convolutional kernels (denoted by red cubes in Fig. 2). The result of this local weighted sum is then passed through a nonlinearity operation such as a rectified linear unit (ReLU). All units in a feature map share the same convolutional kernels. Different feature maps in a convolutional layer usually use different convolutional kernels.

*2) Pooling Layers:* Generally, after each convolutional layer, there exists a pooling layer that is created by computing some local nonlinear operation over a region of the feature maps. The pooling layers aim to reduce the dimension of the representation and create invariance to small translations or rotations, which is very important for image classification and object detection. A widely used pooling operation is max-pooling, which computes the maximum of a local patch of units in one feature map.

*3) FC Layers:* They are typically used as the last few layers of the network to better summarize the information conveyed by lower level layers in view of the final decision.

### *B. Metric Learning*

Metric learning focuses on finding an appropriate similarity measurement between pairs of data (e.g., images) that preserves desired distance structure. Most of the existing metric learning works can be categorized into two main streams: contrastive embedding and triplet embedding [60]–[63].

*1) Contrastive Embedding:* Contrastive embedding is trained on paired data $(x_i, x_j)$ using the following cost function:

$$J = \sum_{i,j} \ell_{ij} D^2(x_i, x_j) + (1 - \ell_{ij}) h(\alpha - D(x_i, x_j))^2 \quad (1)$$

where the label indicator $\ell_{ij} \in \{1, 0\}$ indicates if paired data $(x_i, x_j)$ are from the same class or not. $h(x) = \max(0, x)$ is the hinge loss function and $D(x_i, x_j)$ is the Euclidean distance of paired data $(x_i, x_j)$ with the following definition:

$$D(x_i, x_j) = \|f(x_i) - f(x_j)\|_2 \quad (2)$$

where $f(x)$ is the feature embedding output of the sample $x$ and $\|\cdot\|_2$ denotes the L2-norm operation.

The first term of (1) minimizes the similar pair distances. For dissimilar pairs, we hope that their distances are bigger than a margin $\alpha$, so the second term of (1) is used to penalize the dissimilar pair distances for being smaller than a predefined margin $\alpha$ using hinge loss function.

*2) Triplet Embedding:* Triplet embedding is trained on triplet data $(x_a, x_p, x_n)$ using the following cost function:

$$J = \sum_{a,p,n} h(D(x_a, x_p) - D(x_a, x_n) + \alpha)^2. \quad (3)$$

Here, a triplet $(x_a, x_p, x_n)$ is made up of three samples from two different classes that jointly constitute a positive pair and a negative pair using an anchor of $x_a$, where the positive paired data $(x_a, x_p)$ have the same class label and the negative paired data $(x_a, x_n)$ have different class labels. And the distances of a positive pair and a negative pair are computed by

$$D(x_a, x_p) = \|f(x_a) - f(x_p)\|_2 \quad (4)$$

$$D(x_a, x_n) = \|f(x_a) - f(x_n)\|_2. \quad (5)$$

Intuitively, we hope that the negative pair distance is larger than the positive pair distance plus some margin. To achieve this, we use (3) to penalize the negative pair distances for being smaller than positive pair distances plus a pre-defined margin $\alpha$ using hinge loss function.

In this paper, the contrastive embedding is adopted for our D-CNN model training. However, different from most of the existing metric learning works, the metric learning regularization is not only used to learn discriminative feature representations but also explored to train an effective classifier simultaneously.

## IV. Proposed Method

### *A. Overview of the Proposed Method*

Fig. 3 illustrates the core idea of our proposed method. The goal of our method is to learn D-CNNs for scene classification in order to address the challenges of within-class diversity and between-class similarity. Thus, we can extract more powerful CNN features to further improve the performance of the state-of-the-art deep learning methods. To this end, apart from minimizing the cross entropy loss (i.e., the softmax classification error from the final FC layer used for the traditional CNN models), we also impose a metric learning regularization term on the CNN features to enforce the D-CNN models to be more discriminative. Thus, in the D-CNN feature spaces as shown in the bottom of Fig. 3, the images from the same scene class are as close as possible and the images of different classes are as far away as possible.

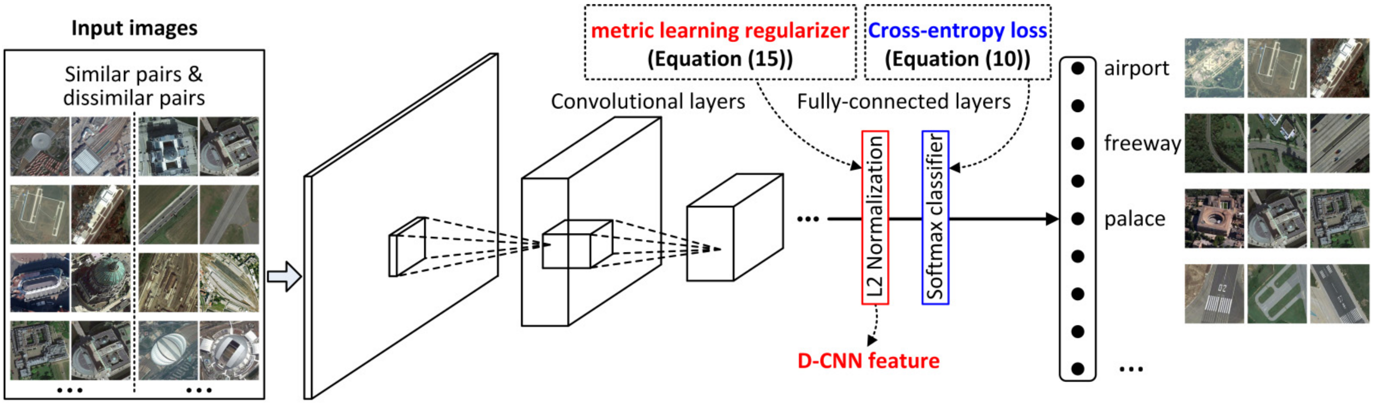


Fig. 3. Illustration of the core idea of the proposed D-CNN method. To address the challenges of within-class diversity and between-class similarity, we propose to learn D-CNNs by optimizing a new objective function. Apart from minimizing the cross-entropy loss, we also impose a metric learning regularization term on the CNN features to enforce the D-CNN models to be more discriminative. Thus, in the D-CNN feature spaces, the images from the same scene class are as close as possible and the images of different classes are as far away as possible.

### *B. Learning Discriminative CNNs*

Let $X = \{x_i | i = 1, 2, \ldots, N\}$ be the set of training samples and $Y = \{\boldsymbol{y}_i | i = 1, 2, \ldots, N\}$ be the set of labels of $X$, where $N$ is the total number of training samples, $\boldsymbol{y}_i \in \mathbb{R}^C$ denotes the ground-truth label vector of sample $x_i$ with only one element being 1 and others being 0, and $C$ is the total number of image scene classes. Suppose our D-CNN model has $L + 1$ layers. For ease of presentation, we simplify the parameters of the network and define $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L, \mathbf{W}_{L+1}\}$ and $\mathbf{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_L, \boldsymbol{B}_{L+1}\}$, where $\mathbf{W}_l$ denotes the filter weights of the $l$th layer and $\boldsymbol{B}_l$ refers to the corresponding biases, $l = 1, \ldots, L + 1$. Thus, the $(L + 1)$th layer is the softmax classification layer and the $L$th layer is the output of our proposed D-CNN feature. Given an input $x_i$, the outputs of the softmax layer $\boldsymbol{O}_{L+1}(x_i)$, the D-CNN feature layer $\boldsymbol{O}_L(x_i)$, and all the other intermediate layers $\boldsymbol{O}_l(x_i)$ are defined as

$$\boldsymbol{O}_{L+1}(x_i) = S_{L+1}(\mathbf{W}_{L+1}\boldsymbol{O}_L(x_i) + \mathbf{B}_{L+1}) \tag{6}$$

$$\boldsymbol{O}_L(x_i) = \frac{S_L(\mathbf{W}_L\boldsymbol{O}_{L-1}(x_i) + \mathbf{B}_L)}{\|S_L(\mathbf{W}_L\boldsymbol{O}_{L-1}(x_i) + \mathbf{B}_L)\|_2} \tag{7}$$

$$\boldsymbol{O}_l(x_i) = S_l(\mathbf{W}_l\boldsymbol{O}_{l-1}(x_i) + \mathbf{B}_l) \tag{8}$$

where $S_l$ is an element-wise nonlinear activation function such as ReLU and softmax. In our work, $S_{L+1}$ is the softmax function. Besides, since we next need to compute the distance in the D-CNN feature space, the D-CNN feature should be L2 normalized to eliminate the scale differences.

As shown in Fig. 3, in order to learn D-CNNs, apart from minimizing the cross-entropy loss, we also impose a metric learning regularization term on the CNN features to enforce the D-CNN models to be more discriminative. To this end, we propose the following new objective function, which consists of three terms including a cross-entropy loss term, a metric learning regularization term, and a weight decay term:

$$J = \min\left(J_1(X, \mathbf{W}, \boldsymbol{B}) + \frac{\lambda_1}{2}J_2(X, \mathbf{W}, \boldsymbol{B}) + \frac{\lambda_2}{2}J_3(\mathbf{W}, \boldsymbol{B})\right) \tag{9}$$

where $\lambda_1$ and $\lambda_2$ are two tradeoff parameters that control the relative importance of these three terms.

*1) Cross-Entropy Loss Term:* This term is defined as the cross entropy loss function as the traditional CNN models. It aims to minimize the classification error for the given training samples and is computed by

$$J_1(X, \mathbf{W}, \boldsymbol{B}) = -\frac{1}{N}\sum_{i=i}^{N}\langle \boldsymbol{y}_i, \log \boldsymbol{O}_{L+1}(x_i)\rangle \tag{10}$$

where $\langle \boldsymbol{y}_i, \log \boldsymbol{O}_{L+1}(x_i)\rangle$ is the inner product of $\boldsymbol{y}_i$ and $\log \boldsymbol{O}_{L+1}(x_i)$, and $N$ is the number of training samples in $X$.

*2) Metric Learning Regularization Term:* This term enforces the D-CNN models to be more discriminative so that the feature representations have small intraclass scatter and big interclass separation. To this end, given each paired training samples $(x_i, x_j)$, their pair-wise feature distance metric can be measured by computing the Euclidean distance between the D-CNN feature representations, which is defined as follows:

$$D(x_i, x_j) = \|\boldsymbol{O}_L(x_i) - \boldsymbol{O}_L(x_j)\|_2. \tag{11}$$

In order to exploit the discriminative feature representations at the output layer of our deep D-CNN model, we expect that the distances between similar pairs are smaller than those between dissimilar pairs and there is a large margin between the similar pairs and dissimilar pairs. In our implementation, the similar and dissimilar pairs are constructed according to the image scene classes. Specifically, if two images share the same scene label, they will be considered a similar pair; otherwise, they will be considered a dissimilar pair. Besides, to prevent the problem of data imbalance, the dissimilar pairs were selected with the same number of similar pairs. To this end, if $x_i$ and $x_j$ are from the same scene class, their feature distance $D(x_i, x_j)$ should be smaller than an up-margin $\tau_1$; if $x_i$ and $x_j$ are from different scene classes, their feature distance $D(x_i, x_j)$ should be bigger than an down-margin $\tau_2$. The formulation can be represented as

$$\begin{cases} D^2(x_i, x_j) < \tau_1, & \boldsymbol{y}_i = \boldsymbol{y}_j \\ D^2(x_i, x_j) > \tau_2, & \boldsymbol{y}_i \neq \boldsymbol{y}_j \end{cases} \tag{12}$$

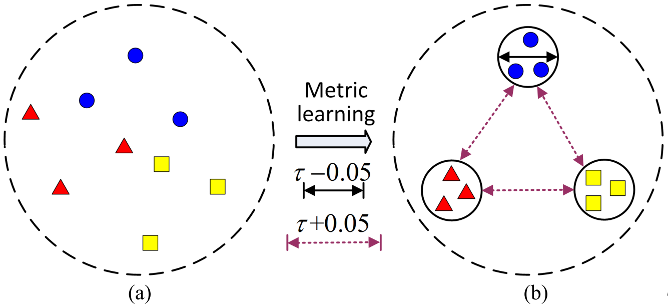


Fig. 4. Illustration of our proposed D-CNN method. By embedding the metric learning constrain to D-CNN model training process, we can obtain a big margin between each similar pair (denoted with the same colors) and dissimilar pair in the learned D-CNN feature space. (a) Original CNN feature space. (b) Our D-CNN feature space.

where $\tau_1$ is used to penalize the similar pair distances, $\tau_2$ is used to constrain the dissimilar pair distances in the training process, and $\tau_2$ should be obviously bigger than $\tau_1$.

To reduce the number of free parameters in the proposed D-CNN model, we introduce an intermediate parameter $\tau$ to connect $\tau_1$ and $\tau_2$ by following the work in [64]. Specifically, by setting $\tau_1 = \tau - 0.05$ and $\tau_2 = \tau + 0.05$, we can simplify the constraint in (12) as follows:

$$0.05 - y_{ij}(\tau - D^2(x_i, x_j)) < 0 \tag{13}$$

where $y_{ij}$ is a label indicator for the paired data $(x_i, x_j)$, which is defined as follows:

$$y_{ij} = \begin{cases} +1, & \boldsymbol{y}_i = \boldsymbol{y}_j \\ -1, & \boldsymbol{y}_i \neq \boldsymbol{y}_j. \end{cases} \tag{14}$$

As shown in Fig. 4, with this constrain in (13), there is a margin between each similar and dissimilar pairs in the learned D-CNN feature space. By applying this constrain to each similar pair and dissimilar pair in the training set, we obtain the hinge loss function of the metric learning regularization term, which is defined as follows:

$$J_2(X, \mathbf{W}, \boldsymbol{B}) = \sum_{i,j} h\left(0.05 - y_{ij}\left(\tau - \|\boldsymbol{O}_L(x_i) - \boldsymbol{O}_L(x_j)\|_2^2\right)\right) \tag{15}$$

where $h(x) = \max(0, x)$ is the hinge loss function. The value of $\tau$ is computed as follows: by assuming that the distances of similar pairs and dissimilar pairs obey normal distribution, we first draw the probability density distribution curves of similar pairs and dissimilar pairs using normal CNN features, and then obtain the junction of these two curves. Here, the $x$-axis value of the junction is the to-be-computed value of $\tau$.

*3) Weight Decay Term:* This term is designed to decrease the magnitudes of the weights of $\mathbf{W}$ and $\boldsymbol{B}$ and hence is useful in preventing over-fitting, which is formulated as follows:

$$J_3(\mathbf{W}, \boldsymbol{B}) = \sum_{l=1}^{L+1} \left(\|\mathbf{W}_l\|_F^2 + \|\boldsymbol{B}_l\|_2^2\right). \tag{16}$$

By incorporating (10), (15), and (16) into (9), we can formulate the training of our proposed D-CNN as the following optimization problem:

$$J = \min \begin{pmatrix} -\frac{1}{N}\sum_{i=i}^{N} \langle \boldsymbol{y}_i, \log \boldsymbol{O}_{L+1}(x_i)\rangle \\ +\frac{\lambda_1}{2}\sum_{i,j} h\left(0.05 - y_{ij}\left(\tau - \|\boldsymbol{O}_L(x_i) - \boldsymbol{O}_L(x_j)\|_2^2\right)\right) \\ +\frac{\lambda_2}{2}\sum_{l=1}^{L+1}\left(\|\mathbf{W}_l\|_F^2 + \|\boldsymbol{B}_l\|_2^2\right) \end{pmatrix}. \tag{17}$$

### C. Optimization

In our work, we solve the optimization problem in (17) using the stochastic gradient descent (SGD) method to obtain the D-CNN neural network parameters of $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L, \mathbf{W}_{L+1}\}$ and $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_L, \boldsymbol{B}_{L+1}\}$. Specifically, the gradients of the objective function $J$ with respect to the parameters $\{\mathbf{W}_{L+1}, \boldsymbol{B}_{L+1}\}$ and $\{\mathbf{W}_l, \boldsymbol{B}_l\}$ $(l = 1, 2, \ldots, L)$ can be computed by

$$\frac{\partial J}{\partial \mathbf{W}_{L+1}} = \frac{1}{N}\sum_{i=1}^{N} \Delta_{L+1}\boldsymbol{O}_L(x_i)^T + \lambda_2 \mathbf{W}_{L+1} \tag{18}$$

$$\frac{\partial J}{\partial \boldsymbol{B}_{L+1}} = \frac{1}{N}\sum_{i=1}^{N} \Delta_{L+1} + \lambda_2 \boldsymbol{B}_{L+1} \tag{19}$$

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{W}_l} &= \frac{1}{N}\sum_{i=1}^{N} \Delta_l \boldsymbol{O}_{l-1}(x_i)^T \\ &\quad + \lambda_1 \sum_{i,j}\left(\Delta_l^{ij}\boldsymbol{O}_{l-1}(x_i)^T + \Delta_l^{ji}\boldsymbol{O}_{l-1}(x_j)^T\right) + \lambda_2 \mathbf{W}_l\end{aligned} \tag{20}$$

$$\frac{\partial J}{\partial \boldsymbol{B}_l} = \frac{1}{N}\sum_{i=1}^{N} \Delta_l + \lambda_1 \sum_{i,j}\left(\Delta_l^{ij}\boldsymbol{O}_{l-1} + \Delta_l^{ji}\boldsymbol{O}_{l-1}\right) + \lambda_2 \boldsymbol{B}_l \tag{21}$$

where $\boldsymbol{O}_0(x_i) = x_i$ and $\boldsymbol{O}_0(x_j) = x_j$, which are the inputs of our D-CNN. The updating equations are computed as

$$\Delta_{L+1} = \boldsymbol{O}_{L+1}(x_i) - \boldsymbol{y}_i \tag{22}$$

$$\Delta_L = \mathbf{W}_{L+1}^T \Delta_{L+1} \odot \mathbf{Z}(\boldsymbol{u}_L(x_i)) S_L'(\boldsymbol{u}_L(x_i)) \tag{23}$$

$$\Delta_l = \mathbf{W}_{l+1}^T \Delta_{l+1} \odot S_l'(\boldsymbol{u}_l(x_i)), \quad l = 1, 2, \ldots, L-1 \tag{24}$$

$$\Delta_L^{ij} = h'(\boldsymbol{v}) y_{ij} (\boldsymbol{O}_L(x_i) - \boldsymbol{O}_L(x_j)) \odot \mathbf{Z}(\boldsymbol{u}_L(x_i)) S_L'(\boldsymbol{u}_L(x_i)) \tag{25}$$

$$\Delta_L^{ji} = h'(\boldsymbol{v}) y_{ij} (\boldsymbol{O}_L(x_j) - \boldsymbol{O}_L(x_i)) \odot \mathbf{Z}(\boldsymbol{u}_L(x_j)) S_L'(\boldsymbol{u}_L(x_j)) \tag{26}$$

$$\Delta_l^{ij} = \mathbf{W}_{l+1}^T \Delta_{l+1}^{ij} \odot S_l'(\boldsymbol{u}_l(x_i)), \quad l = 1, 2, \ldots, L-1 \tag{27}$$

$$\Delta_l^{ji} = \mathbf{W}_{l+1}^T \Delta_{l+1}^{ji} \odot S_l'(\boldsymbol{u}_l(x_j)), \quad l = 1, 2, \ldots, L-1 \tag{28}$$

where the operation $\odot$ denotes element-wise multiplication, and $\mathbf{Z}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ are three intermediate symbols used for simplifying the formulations (23)–(28), which are defined as

$$\mathbf{Z}(\boldsymbol{u}_L(x_i)) \triangleq \frac{\mathbf{I}}{\|S_L(\boldsymbol{u}_L(x_i))\|_2} - \frac{S_L(\boldsymbol{u}_L(x_i))S_L(\boldsymbol{u}_L(x_i))^T}{\|S_L(\boldsymbol{u}_L(x_i))\|_2^3} \tag{29}$$

$$\boldsymbol{u}_l(x_i) \triangleq \mathbf{W}_l\boldsymbol{O}_{l-1}(x_i) + \boldsymbol{B}_l, \quad l = 1, 2, \ldots, L \tag{30}$$

$$\boldsymbol{v} \triangleq 0.05 - y_{ij}\left(\tau - \|\boldsymbol{O}_L(x_i) - \boldsymbol{O}_L(x_j)\|_2^2\right) \tag{31}$$

where $\mathbf{I}$ is the identity matrix.

Thus, the parameters $\{\mathbf{W}_{L+1}, \boldsymbol{B}_{L+1}\}$ and $\{\mathbf{W}_l, \boldsymbol{B}_l\}$ can be updated using the gradient descent method as follows:

$$\begin{aligned}\mathbf{W}_{L+1} &= \mathbf{W}_{L+1} - \mu\frac{\partial J}{\partial \mathbf{W}_{L+1}}\\ \mathbf{W}_l &= \mathbf{W}_l - \mu\frac{\partial J}{\partial \mathbf{W}_l}, \quad l = 1, \ldots, L\end{aligned} \tag{32}$$

$$\begin{aligned}\boldsymbol{B}_{L+1} &= \boldsymbol{B}_{L+1} - \mu\frac{\partial J}{\partial \boldsymbol{B}_{L+1}}\\ \boldsymbol{B}_l &= \boldsymbol{B}_l - \mu\frac{\partial J}{\partial \boldsymbol{B}_l}, \quad l = 1, \ldots, L\end{aligned} \tag{33}$$

where $\mu$ is the learning rate.

## V. Experiments

### A. Data Set Description

In the experiments, we evaluate the proposed D-CNN method on three publicly available data sets designed for remote sensing image scene classification. They are UC Merced data set [65], AID data set [2], and NWPU-RESISC45 data set [1].

The UC Merced data set[1] [65] contains 21 land-use classes including 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. For each scene class, there are 100 aerial images with a size of 256 × 256 pixels and with a spatial resolution of 0.3 m in the red green blue (RGB) color space. This data set has been widely used for the task of remote sensing image scene classification since its emergence.

The Aerial Image data set (AID)[2] [2] is a large-scale data set for aerial scene classification. It contains totally 10 000 images with a size of 600 × 600 pixels within 30 scene classes including airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks, and viaduct. The numbers of images vary from 220 to 420 with different aerial scene classes. The spatial resolution changes from about 8 m to about half a meter.

The NWPU-RESISC45 data set[3] [1] contains totally 31 500 images divided into 45 scene classes. Each class consists of 700 images with a size of 256 × 256 pixels in the RGB color space. The spatial resolution changes from about 30 to 0.2 m per pixel for most of the scene classes. These 45 scene classes include airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, intersection, island, lake, meadow, medium residential, mobile home park, mountain, overpass, palace, parking lot, railway, railway station, rectangular farmland, river, roundabout, runway, sea ice, ship, snowberg, sparse residential, stadium, storage tank, tennis court, terrace, thermal power station, and wetland. To the best of our knowledge, this data set is of the largest scale on the number of scene classes and the total number of images. The rich image variations, large within-class diversity, and high between-class similarity make the data set more challenging.

### B. Data Set Setting and Evaluation Metrics

For the UC Merced data set [65], we randomly split it into 80% for training and 20% for testing. For the AID data set [2], we set the ratios of the number of training set to 20% and 50%, and the left 80% and 50% for testing, while for the NWPU-RESISC45 data set [1], the training ratios are set to 10% and 20%, respectively, and the rest 90% and 80% for testing. These setting are the same as the works in [1], [2], and [65].

Two commonly used metrics including overall accuracy and confusion matrix are adopted to quantitatively evaluate the scene classification results. The overall accuracy is defined as the number of correctly classified images, regardless of which class they belong to, divided by the total number of images. The confusion matrix is an informative table used for analyzing the errors and confusions between different scene classes and it is obtained by counting each class of correct and incorrect classification of the test images and accumulating the results in the table. In addition, to obtain reliable results, on all three data sets, we repeated the experiment 10 times for each training ratio by randomly selecting the training samples and report the mean and standard deviation of the results.

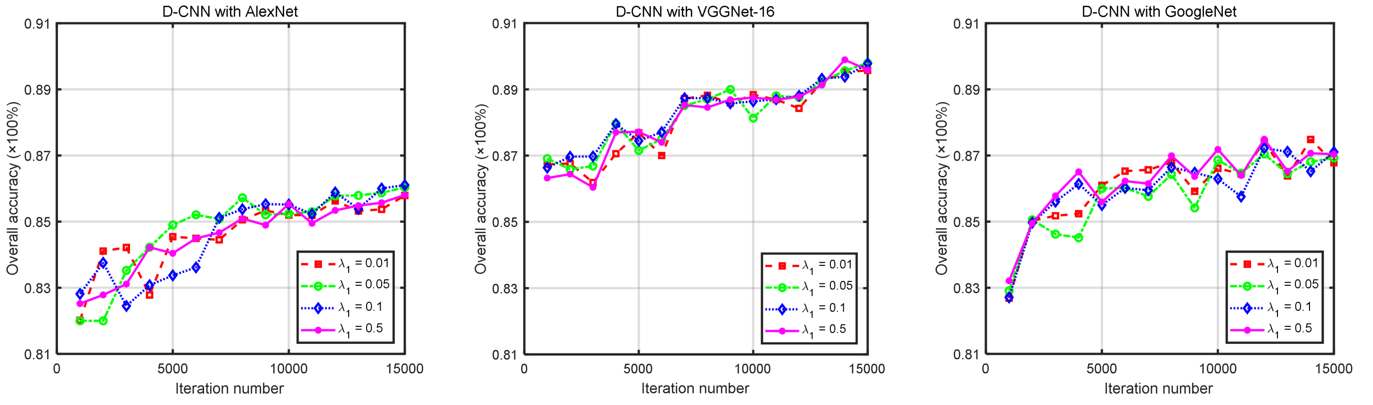
### C. Parameter Setting

We implement our proposed D-CNN method based on three widely used CNN models including AlexNet [6], VGGNet-16 [14], and GoogLeNet [15], which are pretrained on ImageNet. One can refer to [1] for more details about these three CNN models. For our D-CNN model training, the weight decay parameter [i.e., $\lambda_2$ in (17)] is set to 0.0005 and the momentum is set to 0.9. The learning rate is set to 0.01 for the classification layer and 0.001 for the other layers. The intermediate parameter $\tau$ is set to 0.44 for all three data sets. In each SGD iteration, we randomly sample $2(C-1)$ images to construct $(C-1)$ similar pairs and $(C-1)$ dissimilar pairs to obtain a mini-batch of size $2(C-1)$ pairs, where $C$ is the total number of scene classes. Specifically, the $(C-1)$ images are from one same scene class and the other $(C-1)$ images are from the rest $(C-1)$ scene classes. Thus, the similar pairs are obtained by randomly selecting two images from the $(C-1)$ images with the same label and the dissimilar pairs

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

[2] http://www.lmars.whu.edu.cn/xia/AID-project.html

[3] http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html



(a) (b) (c)

Fig. 5. Overall accuracies of our proposed D-CNN methods on the NWPU-RESISC45 data set (10% training ratio) under different parameter settings with three CNN models including (a) AlexNet, (b) VGGNet-16, and (c) GoogLeNet. The metric learning regularization parameter $\lambda_1$ varies from the set of $\{0.01, 0.05, 0.1, 0.5\}$ and the iteration number changes from 1000 to 15 000 with a stride of 1000.

are obtained by randomly selecting two images with different labels.

The metric learning regularization parameter $\lambda_1$ is optimized on the largest scale NWPU-RESISC45 [1] and then kept fixed for the other two data sets. To be specific, we construct the comparison experiments based on three CNN models including AlexNet [6], VGGNet-16 [14], and GoogLeNet [15] by setting the training ratio to 10%, varying the parameter $\lambda_1$ from the set of $\{0.01, 0.05, 0.1, 0.5\}$, and changing the iteration number from 1000 to 15 000 with a stride of 1000, respectively. Fig. 5 reports the detailed results. As can be seen from Fig. 5, our proposed D-CNN method is insensitive to the choice of the regularization parameter $\lambda_1$. Considering the results of all three CNN models, we set the metric learning regularization parameter $\lambda_1$ to 0.05 for our all subsequent experiments.

### *D. Experimental Results and Comparisons*

This paper focuses on enriching the power of the CNN feature representations by training D-CNN models, so we mainly compare our method with CNN feature-based methods including three transferred CNN features and three fine-tuned CNN features. Specifically, for transferred CNN feature-based methods, three off-the-shelf CNN models including AlexNet [6], VGGNet-16 [14], and GoogLeNet [15] are first adopted as universal feature extractors to extract CNN features and then linear one-versus-all support vector machines (SVMs) are used for classification. For fine-tuned CNN feature-based methods, we fine-tune the aforementioned three CNN models on the training data sets to extract better CNN features and then adopt linear one-versus-all SVMs for classification. We implement SVM classification with the LibSVM toolbox and used the default setting in linear SVM. Here, for the transferred CNN features and fine-tuned CNN features, we use linear SVM classifier rather than softmax because SVM classifier could obtain better results, which have been validated by [42]. The experiments were implemented on a workstation with two 2.8-GHz six-core CPUs and 64-GB memory. The transferred CNN feature extraction, CNN model fine-tuning, and our D-CNN model training were implemented with a NVIDIA GTX Titan X GPU for acceleration.

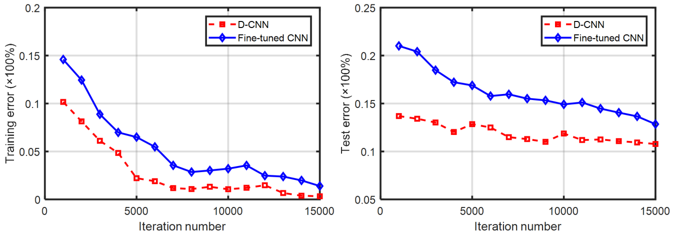


Fig. 6. (Left) Training error and (Right) test error on NWPU-RESISC45 data set (10% training images) with fine-tuned CNN and our proposed D-CNN. Here, VGGNet-16 is the base model and the iteration number changes from 1000 to 15 000 with a stride of 1000.

To validate the effectiveness of our proposed method, Fig. 6 (left) and (right) shows the classification errors of training set and test set on NWPU-RESISC45 data set using 10% training samples with fine-tuned CNN and our proposed D-CNN, respectively. Here, VGGNet-16 is the base model and the iteration number changes from 1000 to 15 000 with a stride of 1000. As can be seen from Fig. 6, our proposed D-CNN outperforms fine-tuned CNN with a big margin on both training set and test set.

Tables I–III present the overall accuracies and standard deviations of the proposed D-CNN methods and six deep feature learning methods, using AlexNet [6], VGGNet-16 [14] and GoogLeNet [15], on the UC Merced data set [65], AID data set [2], and NWPU-RESISC45 data set [1], respectively. The following can be seen from Tables I–III.

1) Using the same CNN model, the transferred CNN feature-based method has the lowest classification accuracy, fine-tuned CNN feature-based method takes the second place, and our D-CNN method has the highest accuracy.
2) The proposed D-CNN method works better than the baselines when the training samples are small. This is because that the embedded metric learning regularization term could obtain more information benefits from

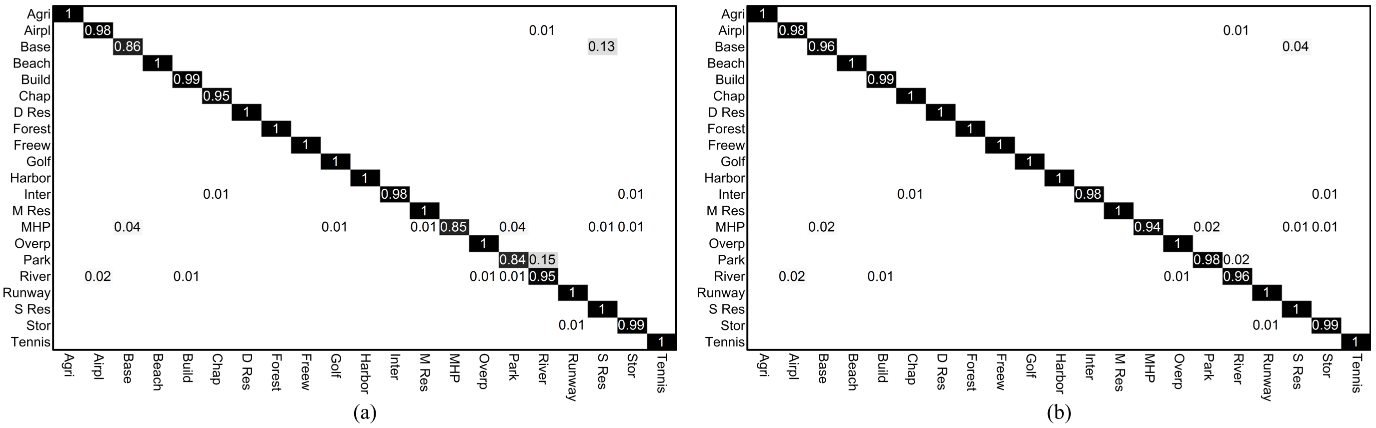


Fig. 7. Confusion matrices of the UC Merced data set under the training ratio of 80% using the following methods. (a) Fine-tuned VGGNet-16 + SVM. (b) D-CNN with VGGNet-16.

TABLE I

Overall Accuracies and Standard Deviations (%) of the Proposed D-CNN Methods and Six Deep Learning Methods on the UC Merced Data Set. The Entry With the Highest Accuracy Is Bold-Faced

| Method | Accuracy |
|---|---|
| Transferred AlexNet + SVM | 94.42±0.10 |
| Fine-tuned AlexNet + SVM | 94.58±0.11 |
| D-CNN with AlexNet (ours) | 96.67±0.10 |
| Transferred GoogLeNet + SVM | 95.32±0.10 |
| Fine-tuned GoogLeNet + SVM | 96.82±0.20 |
| D-CNN with GoogLeNet (ours) | 97.07±0.12 |
| Transferred VGGNet-16 + SVM | 95.24±0.10 |
| Fine-tuned VGGNet-16 + SVM | 97.14±0.10 |
| D-CNN with VGGNet-16 (ours) | **98.93±0.10** |

TABLE II

Overall Accuracies and Standard Deviations (%) of the Proposed D-CNN Methods and Six Deep Learning Methods on the AID Data Set. The Entry with the Highest Accuracy Is Bold-Faced

| Method | Training ratio | |
|---|---|---|
| | 20% | 50% |
| Transferred AlexNet + SVM | 83.22±0.10 | 91.17±0.10 |
| Fine-tuned AlexNet + SVM | 84.23±0.10 | 93.51±0.10 |
| D-CNN with AlexNet (ours) | 85.62±0.10 | 94.47±0.12 |
| Transferred GoogLeNet + SVM | 84.94±0.10 | 92.35±0.10 |
| Fine-tuned GoogLeNet + SVM | 87.51±0.11 | 95.27±0.10 |
| D-CNN with GoogLeNet (ours) | 88.79±0.10 | 96.22±0.10 |
| Transferred VGGNet-16 + SVM | 85.77±0.10 | 93.21±0.10 |
| Fine-tuned VGGNet-16 + SVM | 89.33±0.23 | 96.04±0.13 |
| D-CNN with VGGNet-16 (ours) | **90.82±0.16** | **96.89±0.10** |

TABLE III

Overall Accuracies and Standard Deviations (%) of the Proposed D-CNN Methods and Six Deep Learning Methods on the NWPU-RESISC45 Data Set. The Entry with the Highest Accuracy Is Bold-Faced

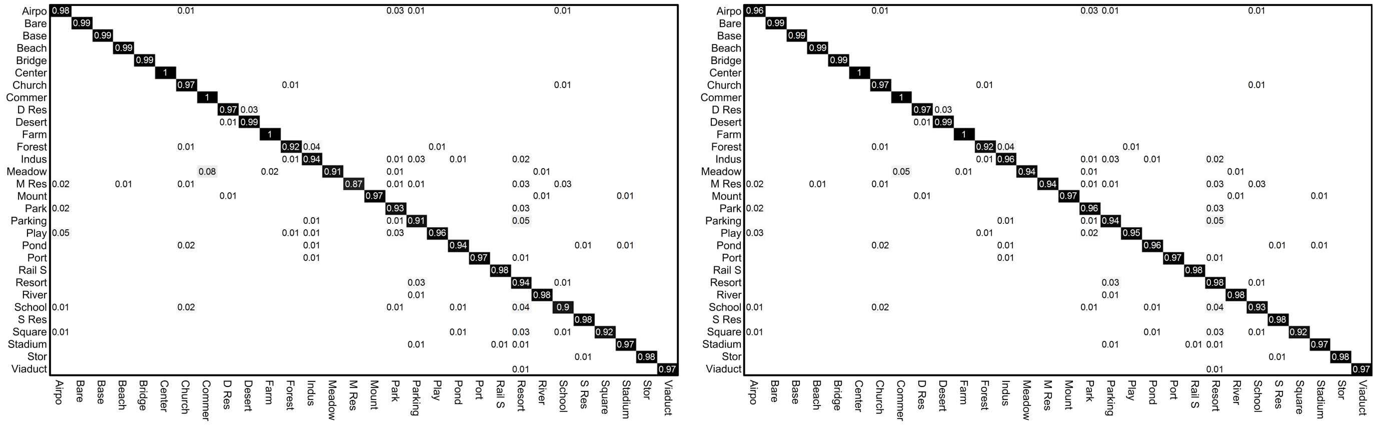
| Method | Training ratio | |
|---|---|---|
| | 10% | 20% |
| Transferred AlexNet + SVM | 76.69±0.21 | 79.85±0.13 |
| Fine-tuned AlexNet + SVM | 81.22±0.19 | 85.16±0.18 |
| D-CNN with AlexNet (ours) | 85.56±0.20 | 87.24±0.12 |
| Transferred GoogLeNet + SVM | 76.19±0.38 | 78.48±0.26 |
| Fine-tuned GoogLeNet + SVM | 82.57±0.12 | 86.02±0.18 |
| D-CNN with GoogLeNet (ours) | 86.89±0.10 | 90.49±0.15 |
| Transferred VGGNet-16 + SVM | 76.47±0.18 | 79.79±0.15 |
| Fine-tuned VGGNet-16 + SVM | 87.15±0.45 | 90.36±0.18 |
| D-CNN with VGGNet-16 (ours) | **89.22±0.50** | **91.89±0.22** |

small-size training data than that from big-size training data.

3) Due to the lack of image variations and diversity of the UC Merced data set [65], its overall accuracy is almost saturated (see Table I) using deep CNN features. While the data sets of AID [2] and NWPU-RESISC45 [1] are still more challenging due to their rich image variations, large within-class diversity, and high between-class similarity.

In addition, we also report the results measured in terms of confusion matrix. Due to the limitation of the space, we present only the confusion matrices under the training ratios of 80% (for UC Merced data set), 50% (for AID data set), and 20% (for NWPU-RESISC45 data set) using our proposed D-CNN method with VGGNet-16 and the fine-tuned VGGNet-16-based method, which are the best and the second best methods for each sets of UC Merced [65], AID [2], and NWPU-RESISC45 [1], respectively, where the entry in the $i$th row and $j$th column denotes the rate of test images from the $i$th class that are classified as the $j$th class. From Figs. 7–9, we observe the following.

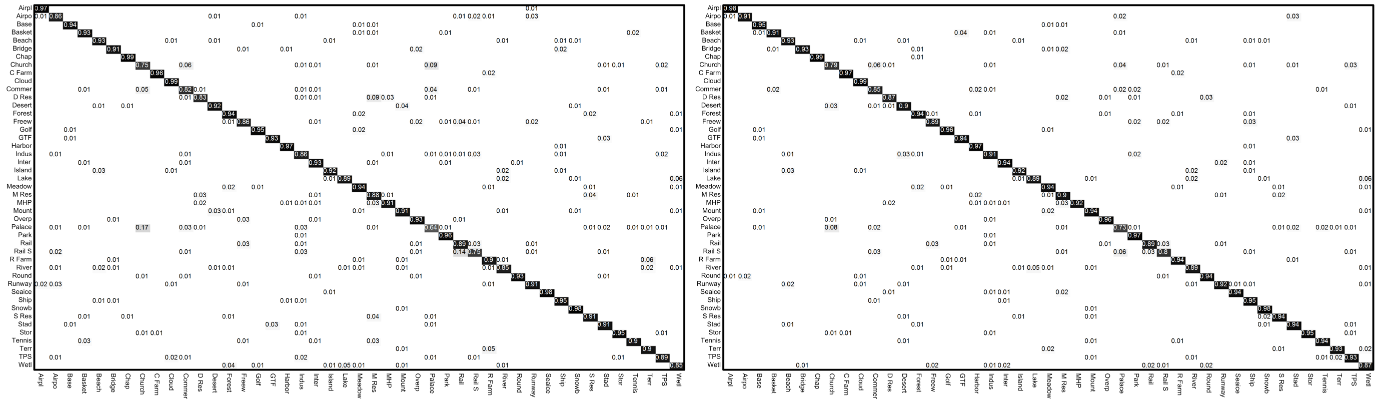
1) We obtain better or at least the same per-class accuracies for 21 out of 21 classes, 28 out of 30 classes, 42 out of 45 classes on UC Merced data set, AID data set, and NWPU-RESISC45 data set, respectively.
2) Using our proposed D-CNN methods, the misclassifications caused by between-class similarity can be



(a)

(b)

Fig. 8. Confusion matrices of the AID data set under the training ratio of 50% using the following methods. (a) Fine-tuned VGGNet-16 + SVM. (b) D-CNN with VGGNet-16.



(a)

(b)

Fig. 9. Confusion matrices of the NWPU-RESISC45 data set under the training ratio of 20% using the following methods. (a) Fine-tuned VGGNet-16 + SVM. (b) D-CNN with VGGNet-16.

significantly reduced. All of these results shown in Tables I–III and Figs. 7–9 show the effectiveness of the proposed method.

## VI. Conclusion

In this paper, we have proposed an effective method to learn D-CNN models for remote sensing image scene classification. Our D-CNN models are trained by optimizing a new discriminative objective function that imposes a metric learning regularization term on the CNN features, apart from minimizing the classification error. In the experiments, the proposed D-CNN methods are comprehensively evaluated on three publicly available data sets based on three widely used CNN models including AlexNet, VGGNet-16, and GoogLeNet. Experimental results demonstrate that our proposed D-CNN methods outperform the existing baseline methods and achieve state-of-the-art results on all three data sets.

## References

[1] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[2] G.-S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[3] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[4] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[5] J. Han *et al.*, "Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding," *ISPRS J. Photogramm. Remote Sens.*, vol. 89, pp. 37–48, Mar. 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[7] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.

[8] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, "Domain adaptation network for cross-scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017.

[9] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.

[10] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[11] E. Li, J. Xia, P. Du, C. Lin, and A. Samat, “Integrating multilayer features of convolutional neural networks for remote sensing scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5653–5665, Oct. 2017.

[12] G. Wang, B. Fan, S. Xiang, and C. Pan, “Aggregating rich hierarchical features for scene classification in remote sensing imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4104–4115, Sep. 2017.

[13] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, “Remote sensing image scene classification using bag of convolutional features,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1735–1739, Oct. 2017.

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–13.

[15] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[16] X. Bian, C. Chen, L. Tian, and Q. Du, “Fusing local and global features for high-resolution scene classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2889–2901, Jun. 2017.

[17] L. Huang, C. Chen, W. Li, and Q. Du, “Remote sensing image scene classification using multi-scale completed local binary patterns and Fisher vectors,” *Remote Sens.*, vol. 8, no. 6, p. 483, 2016.

[18] C. Chen, B. Zhang, H. Su, W. Li, and L. Wang, “Land-use scene classification using multi-scale completed local binary patterns,” *Signal, Image Video Process.*, vol. 10, no. 4, pp. 745–752, 2016.

[19] G. Cheng, P. Zhou, J. Han, L. Guo, and J. Han, “Auto-encoder-based shared mid-level visual dictionary learning for scene classification using very high resolution remote sensing images,” *IET Comput. Vis.*, vol. 9, no. 5, pp. 639–647, Oct. 2015.

[20] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, “Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4238–4249, Aug. 2015.

[21] G. Cheng, J. Han, P. Zhou, and L. Guo, “Multi-class geospatial object detection and geographic image classification based on collection of part detectors,” *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014.

[22] G. Cheng, J. Han, L. Guo, and T. Liu, “Learning coarse-to-fine sparselets for efficient object detection and scene classification,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1173–1181.

[23] J. Zou, W. Li, C. Chen, and Q. Du, “Scene classification using local and global features with collaborative representation fusion,” *Inf. Sci.*, vol. 348, pp. 209–226, Jun. 2016.

[24] X. Lu, X. Li, and L. Mou, “Semi-supervised multitask learning for scene recognition,” *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1967–1976, Sep. 2015.

[25] Y. Wang *et al.*, “Learning a discriminative distance metric with label consistency for scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4427–4440, Aug. 2017.

[26] A. M. Cheriyadat, “Unsupervised feature learning for aerial scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 439–451, Jan. 2014.

[27] G. Sheng, W. Yang, T. Xu, and H. Sun, “High-resolution satellite scene classification using a sparse coding based multiple feature combination,” *Int. J. Remote Sens.*, vol. 33, no. 8, pp. 2395–2412, 2012.

[28] D. Dai and W. Yang, “Satellite image classification via two-layer sparse coding with biased image representation,” *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 1, pp. 173–176, Jan. 2011.

[29] W. Zhou, Z. Shao, C. Diao, and Q. Cheng, “High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder,” *Remote Sens. Lett.*, vol. 6, no. 10, pp. 775–783, 2015.

[30] Y. Li, C. Tao, Y. Tan, K. Shang, and J. Tian, “Unsupervised multilayer feature learning for satellite image scene classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 157–161, Feb. 2016.

[31] F. Zhang, B. Du, and L. Zhang, “Saliency-guided unsupervised feature learning for scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[32] A. Romero, C. Gatta, and G. Camps-Valls, “Unsupervised deep feature extraction for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[33] M. L. Mekhalfi, F. Melgani, Y. Bazi, and N. Alajlan, “Land-use classification with compressive sensing multifeature fusion,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2155–2159, Oct. 2015.

[34] X. Lu, X. Zheng, and Y. Yuan, “Remote sensing scene classification by unsupervised representation learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[35] J. Fan, T. Chen, and S. Lu, “Unsupervised feature learning for land-use scene recognition,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2250–2261, Apr. 2017.

[36] Y. Xu, Z. Wu, J. Li, A. Plaza, and Z. Wei, “Anomaly detection in hyperspectral images based on low-rank and sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 1990–2000, Apr. 2016.

[37] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, “Semantic annotation of high-resolution satellite images via weakly supervised learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[38] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[39] W. Zhao and S. Du, “Scene classification using multi-scale deeply described visual words,” *Int. J. Remote Sens.*, vol. 37, no. 17, pp. 4119–4131, 2016.

[40] F. P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj, “Multiview deep learning for land-use classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2448–2452, Dec. 2015.

[41] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, “Deep learning earth observation classification using ImageNet pretrained networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.

[42] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, “Towards better exploiting convolutional neural networks for remote sensing scene classification,” *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.

[43] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[44] X. Lu, Y. Yuan, and X. Zheng, “Joint dictionary learning for multispectral change detection,” *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 884–897, Apr. 2017.

[45] Q. Zhu, Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang, “Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 747–751, Jun. 2016.

[46] G. Cheng, L. Guo, T. Zhao, J. Han, H. Li, and J. Fang, “Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA,” *Int. J. Remote Sens.*, vol. 34, no. 1, pp. 45–59, 2013.

[47] K. Qi, H. Wu, C. Shen, and J. Gong, “Land-use scene classification in high-resolution remote sensing images using improved correlatons,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2403–2407, Dec. 2015.

[48] L.-J. Zhao, P. Tang, and L.-Z. Huo, “Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4620–4631, Dec. 2014.

[49] S. Chen and Y. Tian, “Pyramid of spatial relatons for scene-level land use classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1947–1957, Apr. 2015.

[50] B. Zhao, Y. Zhong, and L. Zhang, “A spectral–structural bag-of-features scene classifier for very high spatial resolution remote sensing imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 73–85, Jun. 2016.

[51] Z. Wu *et al.*, “GPU parallel implementation of spatially adaptive hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, to be published, doi: 10.1109/JSTARS.2017.2755639.

[52] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao, and Z. Wei, “Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2270–2278, Jun. 2016.

[53] R. Hou, C. Chen, and M. Shah, “Tube convolutional neural network (T-CNN) for action detection in videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5822–5831.

[54] X. Yao, J. Han, D. Zhang, and F. Nie, “Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3196–3209, Jul. 2017.

[55] X. Lu, Y. Chen, and X. Li, “Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features,” *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 106–120, Jan. 2018.

[56] K. Li, G. Cheng, S. Bu, and X. You, “Rotation-insensitive and context augmented object detection in remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, Apr. 2018, doi: 10.1109/TGRS.2017.2778300.

[57] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, “CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion,” *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2761775.

[58] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, "Revealing event saliency in unconstrained video collection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1746–1758, Apr. 2017.

[59] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.

[60] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4004–4012.

[61] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 3608–3616.

[62] G. Cheng, P. Zhou, and J. Han, "Duplex metric learning for image set classification," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 281–292, Jan. 2018.

[63] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2017.2706264.

[64] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1875–1882.

[65] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2010, pp. 270–279.

**Gong Cheng** received the B.S. degree from Xidian University, Xi'an, China, in 2007, and the M.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, in 2010 and 2013, respectively.

He is currently an Associate Professor with Northwestern Polytechnical University. His research interests include computer vision, pattern recognition, and remote sensing image analysis.

**Ceyuan Yang** is currently pursuing the bachelor's degree with Northwestern Polytechnical University, Xi'an, China.

He is currently an Intern with SenseTime Group, Ltd., Beijing, China. His goal is to find solutions to real-world problems using computer vision techniques.

**Xiwen Yao** received the B.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2010 and 2016, respectively.

He is currently a Research Assistant with Northwestern Polytechnical University. His research interests include computer vision and remote sensing image processing, especially on object detection and scene classification.

**Lei Guo** received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 1982 and 1986, respectively, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an, in 1993.

He is currently a Professor with the School of Automation, Northwestern Polytechnical University. His research interest includes image processing.

**Junwei Han** received the B.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1999 and 2003, respectively.

He is currently a Professor with Northwestern Polytechnical University. His research interests include computer vision and multimedia processing.