Tensor Canonical Correlation Analysis Networks for Multi-view Remote Sensing Scene Recognition

Xinghao Yang, Weifeng Liu, *Senior Member, IEEE,* Wei Liu, *Member, IEEE,*

Abstract—Convolutional neural network (CNN) has been proven an effective way to extract high-level features from remote sensing (RS) images automatically. Many variants of the CNN model have been proposed, including principal component analysis network (PCANet), canonical correlation analysis network (CCANet), multiple scale CCANet (MS-CCANet) and multiview CCANet (MCCANet). The PCANet is specialized for single view feature abstraction, while in many real-world practices, the RS data are frequently observed from many more views. Although CCANet, MS-CCANet and MCCANet can be applied to two or more view data, they consider only the pair-wise correlation by calculating a series of *two-order* covariance matrices. However, the high-order consistence, which can only be explored by collectively and simultaneously examining all views, remains undiscovered. In this paper, we propose the tensor canonical correlation analysis network (TCCANet) to tackle this problem. Particularly, TCCANet learns filter banks by simultaneously maximizing arbitrary number of views with high-order-correlation and solves the optimization problem by decomposing a covariance tensor. After the convolutional stage, we utilize binarization and block-wise histogram strategies to generate the final feature. Furthermore, we also develop a Multiple Scale version of TCCANet, i.e., MS-TCCANet, to extract enriched representation of the RS data by incorporating all previous convolutional layers. Numerical experiment results on RSSCN7 and SAT-6 datasets demonstrate the advantages of TCCANet and MS-TCCANet for RS scene recognition.

Index Terms—Tensor canonical correlation analysis, convolutional neural network, multiview learning, remote sensing.

1 Introduction

The advance of satellite and sensor technology brings a number of remote sensing (RS) images and provides opportunities to better understanding the earth surface [1]. RS scene recognition is a main technique in many earth observation tasks, such as, weather reporting [2], [3], military defense [4], [5], traffic monitoring [6], [7] and forest protection [1], [8], and thus attracted widespread attention in the geoscience and remote sensing community. In real-world application, the same region is usually observed via multiple views for better understanding, such as multi-scales [9], multi-angles [10], multi-sensors [11], multi-features [12] and even different seasons [13], as shown in Fig 1. These heterogeneous images provide richer information than single view input, but they also increase the intra-class variability and pose challenges to RS scene recognition technique. So a natural question to ask is how to find out their underlying *consistence features* without losing the *complementary information*?

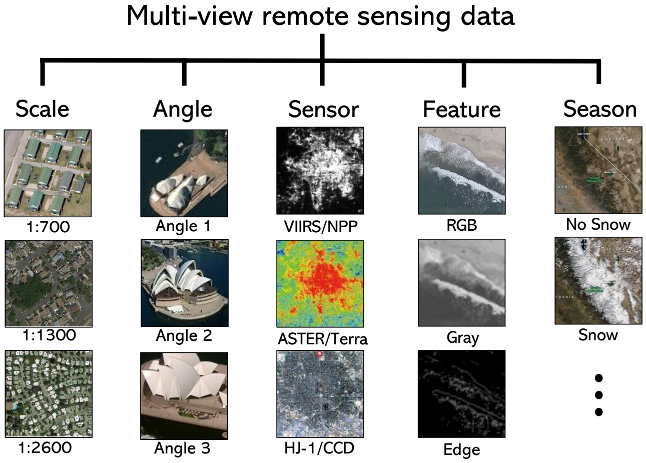


Fig. 1. The multi-view RS data can be collected from different scale (residential area), space angle (Sydney Opera House), sensor/camera (Beijing), data feature (seaside), season (California's Sierra Nevada) and so on. Multi-view RS data exhibit great difference on appearance, even though they depict the same region.

Currently, several multiview RS image classification algorithms have been proposed. For example, Pacifici *et al.* [14] demonstrated that including the acquisition angular information as an additional dimensionality improves the performance of high-resolution RS imagery analysis from solely depending on the pixel digital numbers. Chen *et al.* [15] proposed a deep brief network (DBN) with three parts: the 1-D vector input layer, three hidden restricted Boltzmann machine (RBM), and the logistic regression layer, where the input layer stacks both spectral and spatial information. Luus *et al.* [16] found that the multiscale input

- *X. Yang and Wei Liu are with the School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. E-mail: Xinghao.Yang@student.uts.edu.au, Wei.Liu@uts.edu.au.*
- *Weifeng Liu is with the School of Information and Control Engineering, China University of Petroleum (East China), Qingdao 266580, China. E-mail: liuwf@upc.edu.cn.*

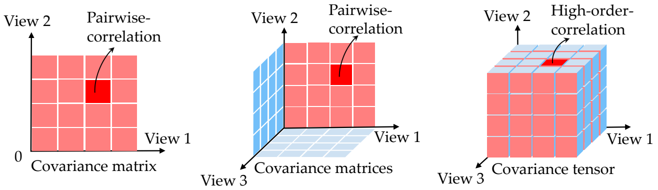
Manuscript received 7 Mar. 2019. (Corresponding author: Weifeng Liu, Wei Liu.)

strategy greatly improves the prediction accuracy of a single CNN in land-use classification compared with the single-scale view input. Similarly, the multiscale strategy is also employed by He *et al.* [17] to enhance the spatial information of hyperspectral image (HSI). Then the handcrafted multiscale covariance maps are fed into a classical 2D-CNN for final classification. Notably, the above methods [14], [15], [16], [17] put multiview data into a single-view feature extraction model with a simple combination (i.e., direct concatenation [14], [15], [16] or load them one by one [17]) before data input, without fully exploring the complementary information and the consensus information hidden in the multiview data [18].

To discover the *complementary information*, Xu *et al.* [19] proposed a two-branch CNN for multisource RS data classification, with one branch extracting the HSI feature and the other branch processing the visible images (VIS). Merging these two view features yields a higher accuracy than using any single one of them. Recently, Li *et al.* [20] designed a two-stage adaptive multiscale deep fusion residual network network (AMDF-ResNet), where the first stage generates multiscale hierarchy features via three residual blocks and the second stage fuses network to select features via different weights. Zhang *et al.* [21] presented the recursive view elimination (RVE) to fuse social sensing data and remote sensing data, aiming to identify risky traffic locations of New York. The social sensing data as the auxiliary view make up for the lack of information in remote sensing images. In [19], [20], [21], the multiview data features are extracted by respective channels of a multiview model so that the complementary information are mostly contained. However, they are powerless in finding the consensus feature, which is really important in reducing the within-class distance (as shown in Fig 1).

In order to excavate the *consistency information* of multi-view data, Wang *et al.* [22] presented the Multiview-based Parameter Free framework (MPF) that explores the coherent property of multi-view subgraphs with a tightness-based merging strategy. Yang *et al.* [23] proposed a two-branch canonical correlation analysis networks (CCANet), with each branch containing eight channels. In CCANet, the canonical correlation analysis (CCA) is employed to calculate the maximally correlated filter kernels for two-view input, so their consistency features are naturally preserved during convolution process. After convolution, the output layer merges the two-view features together to avoid the loss of complementary properties. Based on CCANet, several of its advanced counterparts have been proposed, e.g., the multiple scale CCANet (MS-CCANet) [24] which assembles every convolutional layer's output into the final feature instead of taking only the last convolutional layer, and the multiview CCANet (MCCANet) [12] which extends the CCANet model from two-branch to multi-branch to include arbitrary (more than two) view data and finds the optimal filters by maximizing the sum of all possible pair-wise correlations. However, the methods [12], [23], [24] take only the pair-wise statistical correlations into consideration by analyzing a series of covariance matrices, so they can only be viewed as two-order feature representation tools.

According to the consensus principle [18], finding the maximal agreement from all distinct views simultaneously is an advance way to promote the performance of multi-view embedding. So when we have $V(\geq 3)$ view data, directly finding the high-order consistency property among all views, instead of the roundabout two-view combination, is more desired. Figure 2 is an intuitive illustration for the motivation of pursuing high-order-correlation. Without loss of generality, the number of views is set to $V = 3$ for any $V > 2$ scenarios. By upgrading the covariance matrices to covariance tensor, a straightforward advantage is that more correlation information can be discovered.



(a) (MS-)CCANet [23], [24] (b) MCCANet [12] (c) TCCANet (our method)

Fig. 2. The intuitive difference between low-order-correlation (a)/(b) and high-order-correlation (c). As shown in subfigure (a) and (b), low-order-correlation is optimized by computing covariance matrices, which contain only the two view correlation information. While (c) extends matrix to tensor so that encodes much more correlation information by directly maximizing all views' consistence.

To explore the multi-view complementary information and the consensus information, and simultaneously explore the high-order statistical relationships, in this paper we propose a novel deep convolutional network called tensor canonical correlation analysis network (TCCANet). Particularly, TCCANet is composed of cascaded convolutional layers and the downstream output layer. Each convolutional layer contains multiple branches with each branch processing a separate view input, and the multi-view filter kernels are learned by decomposing a high-order covariance tensor [25]. In the output layer, the binarization and blockwise histogram procedures are utilized to generate the final feature. Figure 3 elucidates the framework of TCCANet with two convolutional layers.

The main contribution of this paper lies in the following four folds:

- We propose TCCANet to learn RS features in a multi-view strategy. Our TCCANet is capable of handing the data that stem from $V(\geq 3)$ views, making it more adaptive to the practical needs. As more view of data carry more complementary information.
- We explore the multiview consistency features by simultaneously maximizing all views' canonical correlation via analyzing a covariance tensor, which is theoretically incorporated more information than the previous low-order methods, i.e., CCANet, MS-CCANet and MCCANet.
- We also design a multiple scale version of TCCANet (MS-TCCANet), which incorporates all previous-layer's features and includes more multi-scale information in the final feature set.
- We perform extensive experiments for multi-view RS image scene recognition to evaluate the proposed TCCANet and MS-TCCANet on RSSCN7 [26] and SAT-6 [27] database.

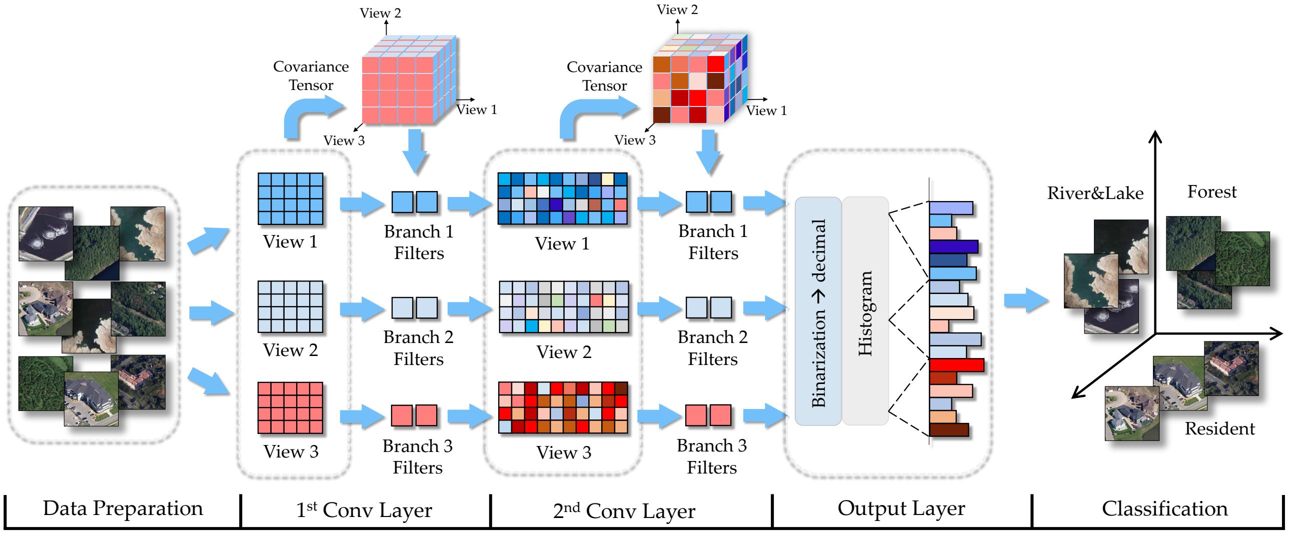


Fig. 3. A general framework of TCCANet. Different color represents view-specific feature. With the convolution going deep, some consistency features are gradually shared. Different branch extracts separate view features, and their complementary information are fused in the output layer. This framework exhibits a two convolutional stages depth network. A deeper structure can be easily constructed by just repeat the former ones.

The rest of this paper is organized as follows: section 2 reviews related work; section 3 formally defines the problem and presents our proposed methods, i.e., TCCANet and MS-TCCANet; experimental results are illustrated in section 4; we conclude this paper in section 5.

2 Related work

In this section, we first review several traditional single-view RS feature learning methods in 2.1 and then briefly formulize several highly related work including both single-view PCANet and multiview CCANet, MS-CCANet and MCCANet in 2.2.

2.1 Single View RS Feature Learning Methods

As RS scene recognition is generally carried out in the feature space, how to construct effective feature representation is a fundamental problem in designing high-performance scene recognition models [28].

The early methods for RS scene classification are largely based on handcrafted features, such as, color histograms [29], local binary patterns (LBPs) [30], scale invariant feature transform (SIFT) [31], histogram of oriented gradients (HOG) [32] and bag of visual words (BoVW) [33]. These features are extracted by manually predefined algorithms with prior domain expertise knowledge. For example, the most popular BoVW [33] employs a pre-computed codebook of visual word (discriminative visual patches) to eliminate the matching time-delay in test images, and has achieved great success for scene classification [34], [35], [36]. However, these handcrafted methods incline to select features for a specific research domain or data type and would lead to poor results on other unknown practical data [37].

In comparison with hand-engineered features that need the involvement of human originality, unsupervised feature learning methods, e.g., principle component analysis (PCA) [38], [39], k-means clustering [40] and sparse coding [41], and deep learning models, e.g., stacked autoencoder (SAE) [42], [43] and convolutional neural networks (CNN) [26], [44], automatically learn features from data itself using a general-purpose learning procedure [45]. For example, Rodarmel and Shan [39] adopted PCA as a data preprocessing technique to select the best spectrum bands for hyperspectral images that are most relevant to classification performance. Nogueira *et al.* [46] exploited the power of six popular CNNs (PatreoNet [47], AlexNet [48], CaffeNet [49], GoogLeNet [50], VGGNet [51] and OverFeat [52]) in RS scene classification under different training strategies, i.e., full training, fine tuning and utilizing CNNs as feature extractors. The results show that the fine-tuned CNNs feature with a linear SVM achieves the best performance. Lu et al. [53] designed the feature aggregation CNN (FACNN) which improves the scene classification accuracy by aggregating CNN's intermediate features in the supervised manner. Wang et al. [54] incorporated the attention mechanism and proposed the attention recurrent convolutional network (ARCNet). It promotes the classification performance by focusing on critical spatial locations and neglecting those trivial features. Chan *et al.* [44] embeded PCA into every convolutional layer to learn multistage filter banks and thus named their method as principle component analysis network (PCANet). The PCANet greatly simplify the traditional CNNs structure by removing the backpropagation procedure. Surprisingly, the naive PCANet with a nearest neighbor (NN) classifier beats both hand-engineered features (e.g., Gabor [55] and LBP [56]) and deep models (e.g., AlexNet [48] and ScatNet [57]) for various kinds of classification applications.

2.2 Comparison of Highly Related Methods

Denote N training images by $\{I_n\}_{n=1}^{N}$ and I_n^v denotes the v^{th} view feature of I_n, with $v = 1, 2, \cdots, V$. Let X^v denote the sample matrix of the v^{th} view, PCANet [44] consists of stacked convolutional layers, and in each convolutional layer, the orthogonal filter banks are found by reshaping the most significant L_1 eigenvectors of X^vX^{vT}

$$W_{l_1} = vec2mat(\mathbf{q}_{l_1}(X^vX^{vT})), l_1 = 1, 2, \cdots, L_1 \tag{1}$$

where W_{l_1} is the ${l_1}^{th}$ filter of the first convolutional layer, $vec2mat(\alpha)$ reshapes a vector $\alpha \in \Re^{k_1k_2}$ to a matrix with size of $k_1 \times k_2$. $\mathbf{q}_{l_1}(X^vX^{vT})$ denotes the ${l_1}^{th}$ primary eigenvector of X^vX^{vT}. In the output layer, the final feature representation is generated within the binarization and block-wise histogram methods.

CCANet [23] extends PCANet from single-view to two-view scenarios by simultaneously finding a common latent subspace for two group data. Particularly, it maximizes the canonical correlation of two-view projected variables as follows

$$\arg\max_{\widetilde{\alpha}} \rho_{12} = corr(z_1, z_2) = \frac{\alpha_1^T S_{12}\alpha_2}{\sqrt{\alpha_1^T S_{11}\alpha_1}\sqrt{\alpha_2^T S_{22}\alpha_2}} \tag{2}$$

where z_1, z_2 and α_1, α_2 are the projected variables and project directions for the two views, respectively. The covariance matrix $S_{ij} = X^iX^{jT}$ and desired canonical vector $\widetilde{\alpha}^T = (\alpha_1^T, \alpha_2^T)$. Then the two-view filter banks can be easily obtained by reshaping the canonical vectors to matrices. Finally, an output layer, followed by the latest convolutional layer, is employed to form the final feature.

MS-CCANet [24] integrates the multi-scale feature from every convolutional layer, instead of only the last one, to fuse more discriminative information. Formally, if $\{f^v_{n,1}\}^2_{v=1}$ and $\{f^v_{n,2}\}^2_{v=1}$ denote the two-view output features of the first and second layers, respectively. Then the final feature of n^{th} sample image can be formulated as

$$f_n = [f^1_{n,1}; f^1_{n,2}; f^2_{n,1}; f^2_{n,2}] \tag{3}$$

Thus, MS-CCANet builds a more fruitful feature by utilizing both two-view's complementary information and different layer's multi-scale feature. However, MS-CCANet is still a two-view representation learning method.

Moreover, MCCANet [12] breaks the two-view limitation and optimizes the multiview filter banks by simultaneously maximizing the sum of all possible pair-wise correlations

$$\begin{aligned} \arg\max_{\widetilde{\alpha}} \sum_{i=1}^{V}\sum_{j=1}^{V}\rho_{ij} &= \sum_{i=1}^{V}\sum_{j=1}^{V} corr(z_i, z_j) \\ &= \sum_{i=1}^{V}\sum_{j=1}^{V}\frac{\alpha_i^T S_{ij}\alpha_j}{\sqrt{\alpha_i^T S_{ii}\alpha_i}\sqrt{\alpha_j^T S_{jj}\alpha_j}} \end{aligned} \tag{4}$$

here $\widetilde{\alpha}^T = (\alpha_1^T, \alpha_2^T, \cdots, \alpha_V^T)$ is employed to form multi-view filter kernels. The final representation is constructed by cascading different view's output into a long vector.

3 Our Method

The TCCANet framework is composed of two modules, i.e., stacked convolutional layers and a subsequent output layer. In each convolutional layer, the multi-view filter kernels are optimized by decomposing a high-order tensor [25]. In the output layer, image binarization and blockwise histogram methods are employed to produce the final representation of input samples. In this section, we first introduce necessary notations and then give the first and second convolutional layers and output layer of TCCANet, respectively. Finally, we extend TCCANet to the multi-scale auxiliary (MS-TCCANet).

3.1 Notations

Let $\mathcal{T}$ be a V-order tensor with dimension of $D_1 \times D_2 \times \cdots \times D_V$, then the Frobenius norm of $\mathcal{T}$ is defined as

$$\|\mathcal{T}\|_F^2 = \sum_{d_1=1}^{D_1}\sum_{d_2=1}^{D_2}\cdots\sum_{d_V=1}^{D_V}\mathcal{T}(d_1, d_2, \cdots, d_V)^2 \tag{5}$$

Let $\mathbf{u}$ be a D_v dimensional vector, the contracted v-mode tensor-vector product is then $\mathcal{A} = \mathcal{T}\bar{\times}_v\mathbf{u}$, where $\mathcal{A}$ is a $V-1$ order tensor with dimension $D_1\times\cdots D_{v-1}\times D_{v+1}\cdots\times D_V$. Formally, $\mathcal{A}$ can be computed as

$$\begin{aligned} &\mathcal{A}(d_1, \cdots, d_{v-1}, d_{v+1}, \cdots, d_V) \\ &= \sum_{d_v=1}^{D_v}\mathcal{T}(d_1, d_2, \cdots, d_V)\mathbf{u}(d_v) \end{aligned} \tag{6}$$

Let U be a $J_v\times D_v$ matrix, the v-mode tensor-matrix product is denoted as $\mathcal{A} = \mathcal{T}\times_v U$, where $\mathcal{A}$ is a $D_1\times\cdots D_{v-1}\times J_v\times D_{v+1}\cdots\times D_V$ tensor where each element

$$\begin{aligned} &\mathcal{A}(d_1, \cdots, d_{v-1}, j_v, d_{v+1}, \cdots, d_V) \\ &= \sum_{d_v=1}^{D_v}\mathcal{T}(d_1, d_2, \cdots, d_V)U(j_v, d_v) \end{aligned} \tag{7}$$

Accordingly, the product between tensor $\mathcal{T}$ and a series of matrices $\{U_v \in \Re^{J_v\times D_v}\}_{v=1}^V$ is a $J_1\times J_2\times\cdots\times J_V$ tensor, which can be expressed as

$$\mathcal{A} = \mathcal{T}\times_1 U_1\times_2 U_2\cdots\times_V U_V \tag{8}$$

This sequence of tensor-matrices product in (8) can be computed by a forward cyclic Kronecker products ($\otimes$):

$$\mathcal{A}_{(v)} = U_v\mathcal{T}_{(v)}(U_{v-1}\cdots\otimes U_1\otimes U_V\otimes\cdots Uv+1) \tag{9}$$

where $\mathcal{T}_{(v)}$ denotes the mode-v unfolding matrix of $\mathcal{T}$ with dimension of $D_v\times(D_1\cdots D_{v-1}D_{v+1}\cdots D_V)$. The mode-$v$ "matrix unfolding" of $\mathcal{T}$ is achieved by mapping the v-fiber as rows and re-arranging all the other fibers as columns one after another.

3.2 The First Convolutional Layer

Suppose we are given a database of N examples, i.e., $\mathcal{S} = \{I_n \in \Re^{w\times h}\}_{n=1}^N$, and the V view features can be denoted as $\mathcal{S}^v = \{I_n^v \in \Re^{w\times h}\}_{n=1}^N, v = 1, 2, \cdots, V$. We first construct a sample matrix for each view by the following steps: (1) select patches around every pixel of I_n^v with size of $k_1\times k_2$, and vectorize them as $x^v_{n,1}, x^v_{n,2}, \cdots, x^v_{n,wh} \in \Re^{k_1k_2}$; (2) organize these vectors into a matrix $X_n^v = [x^v_{n,1}, x^v_{n,2}, \cdots, x^v_{n,wh}] \in \Re^{k_1k_2\times wh}$, and centered as $\bar{X}_n^v$ (i.e., has zero mean); and (3) the sample matrix of the v^{th} view can be denoted as $X^v = [\bar{X}_1^v, \bar{X}_2^v, \cdots, \bar{X}_N^v] = [\mathbf{x}_1^v, \mathbf{x}_2^v, \cdots, \mathbf{x}_{Nwh}^v] \in \Re^{k_1k_2\times Nwh}$, with $\mathbf{x}_i^v$ denotes the i^{th} column of X^v. Then the variance matrix for the v^{th} view is $S_{vv}=X^{v}X^{vT}$ and covariance tensor of all views, i.e., $\mathcal{C}_{12\cdots V}$, could be calculated as

$$\mathcal{C}_{12\cdots V}=\frac{1}{Nwh}\sum_{i=1}^{Nwh}\mathbf{x}_i^1\circ\mathbf{x}_i^2\circ\cdots\circ\mathbf{x}_i^V \tag{10}$$

where $\circ$ is tensor (outer) product and $\mathcal{C}_{12\cdots V}$ is a V-order tensor with each dimension, D_1, D_2, to D_V being k_1k_2.

We learn multi-view filter banks from sample matrices $\{X^v\}_{v=1}^V$ by employing the tensor concept. In contrast to CCANet and MCCANet, where only the pair-wise statistical correlations are considered, TCCANet explores high-order-correlation by directly maximizing the correlation between all canonical variables $z_v=X^{vT}\alpha_v, v=1,2,\cdots,V$. Thus, the optimization problem can be formulated as

$$\begin{aligned}\arg\max_{\widetilde{\alpha}}\rho_{12\cdots V}&=corr(z_1,z_2,\cdots,z_V)\\ s.t.\ z_v^Tz_v&=1,v=1,2,\cdots,V\end{aligned} \tag{11}$$

where $corr(z_1,z_2,\cdots,z_V)=(z_1\odot z_2\odot\cdots\odot z_V)^T\mathbf{e}$ is the high-order canonical correlation, the symbol $\odot$ denotes the element-wise product, and $\mathbf{e}\in\Re^{Nwh}$ represents an all ones column vector. We introduce the following theorem.

Theorem 1. *The high order canonical correlation is given by*

$$\begin{aligned}\rho_{12\cdots V}&=(z_1\odot z_2\odot\cdots\odot z_V)^T\mathbf{e}\\ &=\mathcal{C}_{12\cdots V}\bar{\times}_1\alpha_1^T\bar{\times}_2\alpha_2^T\cdots\bar{\times}_V\alpha_V^T\end{aligned} \tag{12}$$

The proof can be found in Appendix A. By simultaneously considering Theorem 1 and variance matrices $S_{vv}=X^vX^{vT}, v=1,2,\cdots,V$, the original model (11) can be converted to the following problem

$$\begin{aligned}\arg\max_{\widetilde{\alpha}}\rho_{12\cdots V}&=\mathcal{C}_{12\cdots V}\bar{\times}_1\alpha_1^T\bar{\times}_2\alpha_2^T\cdots\bar{\times}_V\alpha_V^T\\ s.t.\ \alpha_v^TS_{vv}\alpha_v&=1,v=1,2,\cdots,V\end{aligned} \tag{13}$$

We then add a regularization item for each constraint, i.e., $\widetilde{S}_{vv}=S_{vv}+\epsilon E$, where ϵ is a non-negative parameter and E is an identity matrix. Therefore, the constraints of model (13) become $\alpha_v^T\widetilde{S}_{vv}\alpha_v=1,v=1,2,\cdots,V$. Let $h_v=\widetilde{S}_{vv}^{\frac{1}{2}}\alpha_v$ and $\mathcal{K}=\mathcal{C}_{12\cdots V}\times_1\widetilde{S}_{11}^{-\frac{1}{2}}\times_2\widetilde{S}_{22}^{-\frac{1}{2}}\cdots\times_V\widetilde{S}_{VV}^{-\frac{1}{2}}$. Then Appendix B proves that the problem (13) is equivalent to (14).

$$\begin{aligned}\arg\max_{\widetilde{h}}\rho_{12\cdots V}&=\mathcal{K}\bar{\times}_1h_1^T\bar{\times}_2h_2^T\cdots\bar{\times}_Vh_V^T\\ s.t.h_v^Th_v&=1,v=1,2,\cdots,V\end{aligned} \tag{14}$$

Lathauwer *et al.* [58] proves that the problem (14) equals to finding the best rank-1 approximation of $\mathcal{K}$. Specifically, the best rank-1 approximation problem minimizes the distance between $\mathcal{K}$ and its rank-1 manifold approximation, i.e., $\hat{\mathcal{K}}\overset{def}{=}\lambda h_1\circ h_2\circ\cdots\circ h_V$, as

$$\arg\min_{\widetilde{h}}\|\mathcal{K}-\hat{\mathcal{K}}\|_F^2 \tag{15}$$

Solving approximation problem (15) by alternating least squares (ALS) algorithm [59] yields the main solution of TCCA. Inspired by CCA, the remaining solutions are obtained by iteratively optimizing the same correlation as presented in (14). This leads to the sum of best rank-1 optimization, a.k.a., the best rank-L CANDECOMP/PARAFAC decomposition [60] of the tensor $\mathcal{K}$

Algorithm 1 The ALS algorithm

```
Input:
    Tensor $\mathcal{K}$;
    The termination threshold $fitchangetol=10^{-4}$;
    The maximal iterative times $maxiters=50$.
Output: Optimized transformation: $h_v, v=1,2,3$.
Randomly initialize $h_v(0)\in\Re^{k_1k_2\times L}, v=1,2,3$;
Initialize $fit=0$;
for $i=1:maxiters$ do
    $fitold=fit$;
    $h_2(i)=f_{h_2}\big(h_3(i-1),h_1(i-1)\big)$;
    $h_3(i)=f_{h_3}\big(h_1(i-1),h_2(i)\big)$;
    $h_1(i)=f_{h_1}\big(h_2(i),h_3(i)\big)$;
    $\mathcal{P}=\sum_{l=1}^{L}\lambda_l(i)h_{1,l}(i)\circ h_{2,l}(i)\circ h_{3,l}(i)$;
    $fit=1-\frac{\|\mathcal{K}-\mathcal{P}\|_F}{\|\mathcal{K}\|_F}$;
    $fitchange=|fitold-fit|$;
    if $fitchange<fitchangetol$ then
        break
    end if
end for
Output transformations: $h_v, v=1,2,3$.
```

$$\mathcal{K}\approx\sum_{l=1}^{L}\lambda_lh_{1,l}\circ h_{2,l}\circ\cdots\circ h_{V,l} \tag{16}$$

Let $h_v=(h_{v,1},h_{v,2},\cdots,h_{v,L})\in\Re^{k_1k_2\times L}$ be the optimal solution for the v^{th} view, where $h_{v,l}$ represents the l^{th} column of h_v. Without loss of generality, we set $V=3$ and formulate the principle of ALS as follows

$$\begin{aligned}h_1^T&=(h_2\diamond h_3)^\dagger\mathcal{K}_{(1)}^T\overset{def}{=}f_{h_1}(h_2,h_3)\\ h_2^T&=(h_3\diamond h_1)^\dagger\mathcal{K}_{(2)}^T\overset{def}{=}f_{h_2}(h_3,h_1)\\ h_3^T&=(h_1\diamond h_2)^\dagger\mathcal{K}_{(3)}^T\overset{def}{=}f_{h_3}(h_1,h_2)\end{aligned} \tag{17}$$

where $\diamond$ is the Khatri-Rao product [59] and $h^\dagger$ is the pseudo-inverse of h. Then h_v is updated by the iterative ALS algorithm, which is illustrated in Algorithm 1. Based on the h_v, we compute the transformation directions of first layer $\alpha_v=\widetilde{S}_{vv}^{-\frac{1}{2}}h_v=[\alpha_{v,1},\alpha_{v,2},\cdots,\alpha_{v,L_1}]\in\Re^{k_1k_2\times L_1}$ which are ordered by the significance of λ_l.

The V-view filter kernels of the first layer, i.e., $\{W_{v,l_1},l_1=1,2,\cdots,L_1\}_{v=1}^V$ are constructed by just reshaping the transformation directions, i.e., $\{\alpha_{v,l_1},l_1=1,2,\cdots,L_1\}_{v=1}^V$, from vectors to matrices

$$W_{v,l_1}=vec2mat(\alpha_{v,l_1})\in\Re^{k_1\times k_2},l_1=1,2,\cdots,L_1 \tag{18}$$

where the function $vec2mat(\alpha)$ maps a vector $\alpha\in\Re^{k_1k_2}$ to a matrix. The parameter L_1 denotes the filter number of first convolutional layer. For each input image I_n^v, it yields L_1 output features when passing by the convolutional stage

$$O_{n,l_1}^v=I_n^v*W_{v,l_1},l_1=1,2,\cdots,L_1. \tag{19}$$

where $*$ is the 2-dimensional discrete convolution operator. It is worth mentioned that the edge of I_n^v should be zero

Algorithm 2 The filter banks optimization algorithm

```
Input: $V$ view sample matrices $X^v$
Output: $V$ view transformation directions $\alpha_v$
1: Compute the covariance tensor $\mathcal{C}_{12\cdots V} = \sum_i \mathbf{x}_i^1 \circ \mathbf{x}_i^2 \circ \cdots \circ \mathbf{x}_i^V$;
2: Calculate variance matrices $S_{vv} = X^v X^{vT}$;
3: Add regularizer item $\widetilde{S}_{vv} = S_{vv} + \epsilon E$;
4: Compute $\mathcal{K} = \mathcal{C}_{12\cdots V} \times_1 \widetilde{S}_{11}^{-\frac{1}{2}} \times_2 \widetilde{S}_{22}^{-\frac{1}{2}} \cdots \times_V \widetilde{S}_{VV}^{-\frac{1}{2}}$;
5: Find the best rank-$L$ approximation of $\mathcal{K}$ by ALS Algorithm 1, i.e., $h_v = (h_{v,1}, h_{v,2}, \cdots, h_{v,L})$;
6: Output $\alpha_v = \widetilde{S}_{vv}^{-\frac{1}{2}} h_v$;
```

padded before the convolutional stage, which is used to ensure its size is the same as that of the input image. For quick reference, we conclude the overall process of filter optimization in Algorithm 2

3.3 The Second Convolutional Layer

The output feature images of first convolutional layer, i.e., $\{O_{n,l_1}^v\}_{l_1=1}^{L_1}, n = 1, 2, \cdots, N$, are fed as the inputs of the second layer. After patch selection, vectorizing and mean removing as described previously, we construct the sample matrix of v^{th} view as $Y^v = [\mathbf{y}_1^v, \mathbf{y}_2^v, \cdots, \mathbf{y}_{NL_1wh}^v] \in \Re^{k_1k_2 \times NL_1wh}$. Then the variance matrices for the second layer are

$$\mathbf{S}_{vv} = Y^v Y^{vT}, v = 1, 2, \cdots, V \tag{20}$$

and the covariance tensor is

$$\mathcal{D}_{12\cdots V} = \frac{1}{NL_1wh} \sum_{i=1}^{NL_1wh} \mathbf{y}_i^1 \circ \mathbf{y}_i^2 \circ \cdots \circ \mathbf{y}_i^V \tag{21}$$

The multiview's consistence filter banks are learned by the optimization Algorithm 2 but with the new input Y^v. Let the transformation directions be $\{\beta_v\}_{v=1}^V$ and the rotated directions $\{\mathbf{h}_v = \widetilde{\mathbf{S}}_{vv}^{\frac{1}{2}} \beta_v\}_{v=1}^V$ with $\widetilde{\mathbf{S}}_{vv} = \mathbf{S}_{vv} + \epsilon E$. Then we discover the high-order-correlation by solving the following problem:

$$\begin{aligned} \arg\max_{\widetilde{\mathbf{h}}} \rho_{12\cdots V} = \mathcal{R} \bar{\times}_1 \mathbf{h}_1^T \bar{\times}_2 \mathbf{h}_2^T \cdots \bar{\times}_V \mathbf{h}_V^T \\ s.t.\ \mathbf{h}_v^T \mathbf{h}_v = 1, v = 1, 2, \cdots, V \end{aligned} \tag{22}$$

where $\mathcal{R} = \mathcal{D}_{12\cdots V} \times_1 \widetilde{\mathbf{S}}_{11}^{-\frac{1}{2}} \times_2 \widetilde{\mathbf{S}}_{22}^{-\frac{1}{2}} \cdots \times_V \widetilde{\mathbf{S}}_{VV}^{-\frac{1}{2}}$. This problem is then extended to a sum of rank-1 problem to acquire a series of directions

$$\mathcal{R} \approx \sum_{l=1}^{L_2} \lambda_l \mathbf{h}_{1,l} \circ \mathbf{h}_{2,l} \circ \cdots \circ \mathbf{h}_{V,l} \tag{23}$$

Solving this problem by the ALS algorithm, we have L_2 directions for each view $\{\beta_v = \widetilde{\mathbf{S}}_{vv}^{-\frac{1}{2}} \mathbf{h}_v = [\beta_{v,1}, \beta_{v,2}, \cdots, \beta_{v,L_2}] \in \Re^{k_1k_2 \times L_2}\}_{v=1}^V$. So the filter banks of the second layer are given by

$$\mathbf{W}_{v,l_2} = vec2mat(\beta_{v,l_2}) \in \Re^{k_1 \times k_2}, l_2 = 1, 2, \cdots, L_2 \tag{24}$$

where $\mathbf{W}_{v,l_2}$ denotes the ${l_2}^{th}$ filter of the v^{th} view. For each input image O_{n,l_1}^v, we obtain L_2 outputs by convolution with L_2 filters

$$\mathbf{O}_{n,l_1l_2}^v = \{O_{n,l_1}^v * \mathbf{W}_{v,l_2}\}_{l_2=1}^{L_2}. \tag{25}$$

This completes the second convolutional layer. Clearly, for each training image I_n, the first convolutional layer produces $V \times L_1$ feature images (i.e., $\{O_{n,l_1}^v\}_{l_1=1}^{L_1}$) and the second convolutional layer yields $V \times L_1 \times L_2$ feature images (i.e., $\{\mathbf{O}_{n,l_1l_2}^v\}_{l_1=1,l_2=1}^{L_1,L_2}$). The proposed TCCANet framework can be extended to a deeper convolutional network by just repeating the former procedure. For simplicity, we only introduce the two-convolutional-layer model.

3.4 The Output Layer

The objective of the output layer is to form the final feature representation $\mathbf{f}_n$ for each training sample I_n. To this end, it implements binarization and blockwise histogram as the nonlinear processing and feature pooling, respectively. In the nonlinear processing stage, we binarize the feature images of the last convolutional layer as $H\Big(\{\mathbf{O}_{n,l_1l_2}^v\}_{l_1=1,l_2=1}^{L_1,L_2}\Big)$. The $H(\cdot)$ is a hashing function, which maps a real-valued number τ as follows

$$H(\tau) = \begin{cases} 1, & \tau > 0 \\ 0, & \tau \leq 0 \end{cases} \tag{26}$$

Therefore, we have $V \times L_1 \times L_2$ binary feature images. For each fixed l_1, we sum the L_2 binary images $\{\mathbf{O}_{n,l_1l_2}^v\}_{l_2=1}^{L_2}$ to one decimal image by their bits weight

$$\mathbf{D}_{n,l_1}^v = \sum_{l_2=1}^{L_2} 2^{l_2-1} H(\mathbf{O}_{n,l_1l_2}^v) \tag{27}$$

where each $\mathbf{D}_{n,l_1}^v$ denotes a decimal image in which every pixel belongs to $[0, 2^{L_2} - 1]$.

In the feature pooling stage, we first partition every decimal image $\mathbf{D}_{n,l_1}^v$ into B blocks and the block size is $b_1 \times b_2$. Secondly, we statistic the histogram of decimal pixels for each block and concatenate these histograms into one vector as $BlkHist(\mathbf{D}_{n,l_1}^v) \in \Re^{2^{L_2}B}$. By repeating this encoding procedure, the histogram feature of I_n^v can be computed as

$$\mathbf{f}_n^v = [BlkHist(\mathbf{D}_{n,1}^v); \cdots; BlkHist(\mathbf{D}_{n,L_1}^v)] \in \Re^{2^{L_2}L_1B} \tag{28}$$

Finally, the final feature representation for each training sample I_n is defined as

$$\mathbf{f}_n = [\mathbf{f}_n^1; \mathbf{f}_n^2; \cdots; \mathbf{f}_n^V] \in \Re^{2^{L_2}VL_1B} \tag{29}$$

The workflow of TCCANet is summarized in Algorithm 3.

3.5 Multiple Scale TCCANet (MS-TCCANet)

In the TCCANet framework, the output layer follows only the last convolutional layer. As a consequence, it extracts the feature only from the last convolutional layer and ignores all the previous layers. In this section, we extend the TCCANet to a multiple scale version, i.e., MS-TCCANet. MS-TCCANet adds an output layer after every convolutional layer, following the idea that different layer contains different discriminative information. In terms of the two-convolutional-layer TCCANet model, MS-TCCANet adds an additional output layer to the end of the first convolutional layer. Specifically, the feature images of the first

Algorithm 3 The TCCANet algorithm

```
Input: $N$ training samples $\mathcal{S} = \{I_n\}_{n=1}^{N}$
Output: $N$ feature vectors $\{\mathbf{f}_n\}_{n=1}^{N}$
Extract $V$ view features $\mathcal{S}^v = \{I_n^v\}_{v=1}^{V}$;
for the first convolutional layer do
    Construct $V$ view sample matrices $\{X^v\}_{v=1}^{V}$;
    Compute $V$ view transformation directions via Algorithm 2, i.e., $\alpha_v = [\alpha_{v,1}, \alpha_{v,2}, \cdots, \alpha_{v,L_1}]$;
    Form filter banks $W_{v,l_1} = vec2mat(\alpha_{v,l_1})$;
    Calculate the output of first layer $O_{n,l_1}^v = I_n^v * W_{v,l_1}$;
end for
for the second convolutional layer do
    Construct $V$ view sample matrices $\{Y^v\}_{v=1}^{V}$;
    Compute $V$ view transformation directions via Algorithm 2, i.e., $\beta_v = [\beta_{v,1}, \beta_{v,2}, \cdots, \beta_{v,L_2}]$;
    Form filter banks $\mathbf{W}_{v,l_2} = vec2mat(\beta_{v,l_2})$;
    Calculate the output of second layer $\mathbf{O}_{n,l_1 l_2}^v = \{O_{n,l_1}^v * \mathbf{W}_{v,l_2}\}$;
end for
Calculate the binary image $H(\mathbf{O}_{n,l_1 l_2}^v)$;
Map $L_2$ binary images to one decimal image $\mathbf{D}_{n,l_1}^v = \sum_{l_2=1}^{L_2} 2^{l_2-1} H(\mathbf{O}_{n,l_1 l_2}^v)$;
Extract the histogram feature $\mathbf{f}_n^v$;
Output the final feature $\mathbf{f}_n = [\mathbf{f}_n^1; \mathbf{f}_n^2; \cdots; \mathbf{f}_n^V]$
```

layer, i.e., $\{O_{n,l_1}^v\}_{l_1=1}^{L_1}$, are first converted into binary images $H\Big(\{O_{n,l_1}^v\}_{l_1=1}^{L_1}\Big)$. Every set of L_1 binary images are then mapped to one decimal image

$$D_n^v = \sum_{l_1=1}^{L_1} 2^{l_1-1} H(O_{n,l_1}^v) \tag{30}$$

Each pixel of D_n^v is an integer which in the range of $[0, 2^{L_1} - 1]$. To pool the histogram feature, each decimal image D_n^v is segmented to B blocks. Then its block-wise histogram feature can be denoted as $BlkHist(D_n^v) \in \Re^{2^{L_1}B}$. The final feature of the first convolutional layer can be obtained by concatenate the V views histogram vector

$$f_{n,1} = [BlkHist(D_n^1); \cdots; BlkHist(D_n^V)] \in \Re^{2^{L_1}VB} \tag{31}$$

Let the final feature obtained from the output layer of the second convolutional layer be $f_{n,2} = \mathbf{f}_n$. Therefore, MS-TCCANet stacks the output features obtained from all convolutional layers together as

$$f_n = [f_{n,1}; f_{n,2}] \in \Re^{(2^{L_1} + 2^{L_2}L_1)VB} \tag{32}$$

We summarize the overall framework MS-TCCANet in Algorithm 4.

Both the TCCANet and MS-TCCANet involve several model parameters, such as, the filter size (or patch size) $k_1 \times k_2$, filter number of first (L_1) and second (L_2) convolutional layer, the block size $b_1 \times b_2$ and the block overlapping ratio. We found the block overlapping ratio has minor impact on the application result, so we fix it to 0.5 in our experiments and investigate the impact of other parameters in Section 4.

4 Experiments

In this section, we evaluate the proposed TCCANet and MS-TCCANet on RSSCN7 [26] and SAT-6 [27] databases for multi-view remote sensing scene recognition. The linear support vector machine (SVM) classifier with the penalty factor $c = 1$ is adopted for the classification tasks of PCANet, CCANet, MS-CCANet, MCCANet, TCCANet and MS-TCCANet. The regularization parameter ϵ is set to 0.01 in this paper. Section 4.1 introduces the datasets and section 4.2 illustrates the experimental results. We also compare the TCCANet and MS-TCCANet with several deep CNN models, i.e., VGGNet, ResNet and SENet, in section 4.3.

Algorithm 4 The MS-TCCANet algorithm

```
Input: $N$ training samples $\mathcal{S} = \{I_n\}_{n=1}^{N}$
Output: $N$ feature vectors $\{f_n\}_{n=1}^{N}$
Extract $V$ view features $\mathcal{S}^v = \{I_n^v\}_{v=1}^{V}$;
for the first convolutional layer do
    Construct $V$ view sample matrices $\{X^v\}_{v=1}^{V}$;
    Compute $V$ view transformation directions via Algorithm 2, i.e., $\alpha_v = [\alpha_{v,1}, \alpha_{v,2}, \cdots, \alpha_{v,L_1}]$;
    Form filter banks $W_{v,l_1} = vec2mat(\alpha_{v,l_1})$;
    Calculate the output of first layer $O_{n,l_1}^v = I_n^v * W_{v,l_1}$;
    Calculate the binary image $H(O_{n,l_1}^v)$;
    Map $L_1$ binary images to one decimal image $D_n^v = \sum_{l_1=1}^{L_1} 2^{l_1-1} H(O_{n,l_1}^v)$;
    Extract the histogram feature of the first layer $f_{n,1}$;
end for
for the second convolutional layer do
    Construct $V$ view sample matrices $\{Y^v\}_{v=1}^{V}$;
    Compute $V$ view transformation directions via Algorithm 2, i.e., $\beta_v = [\beta_{v,1}, \beta_{v,2}, \cdots, \beta_{v,L_2}]$;
    Form filter banks $\mathbf{W}_{v,l_2} = vec2mat(\beta_{v,l_2})$;
    Calculate the output of second layer $\mathbf{O}_{n,l_1 l_2}^v = \{O_{n,l_1}^v * \mathbf{W}_{v,l_2}\}$;
    Calculate the binary image $H(\mathbf{O}_{n,l_1 l_2}^v)$;
    Map $L_2$ binary images to one decimal image $\mathbf{D}_{n,l_1}^v = \sum_{l_2=1}^{L_2} 2^{l_2-1} H(\mathbf{O}_{n,l_1 l_2}^v)$;
    Extract the histogram feature of the second layer $f_{n,2}$;
end for
Output the final feature $f_n = [f_{n,1}; f_{n,2}]$;
```

4.1 Database Description

The RSSCN7 database [26] contains 2800 RS images with seven typical scene classes, i.e., the grassland, farmland, forest, parking lot, residential area, industrial area, and river&lake. In each class, there are 400 images with size of 400×400. All images are collected by Google Earth from different weather, scales and seasons. Figure 4 illustrates several examples of RSSCN7. The three-view features, including a gray feature, an edge feature and a low-frequency wavelet transform (WT) feature, are extracted, and then all images are resized to 64×64 pixels. Figure 5 gives an example of one selected image and its corresponding three-view features.

The SAT-6 [27] database consists of $405,000$ sample images with size 28×28 and including 6 landcover scene classes — buildings, water bodies, trees, barren land, grassland, and roads. All these images are divided into four-fifths $(324,000)$ for training and the rest $(81,000)$ for testing. In our experiment, a subset of SAT-6 is utilized to evaluate our algorithm. Particularly, the subset is composed of 18000

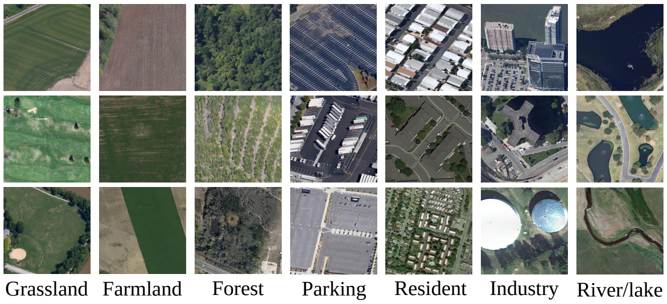


Fig. 4. Several samples of RSSCN7 database. Each column represents a specific class whose label is given in the bottom of each column.

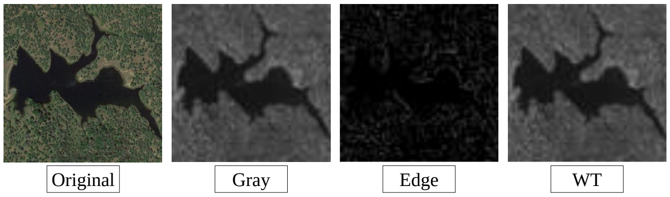


Fig. 5. One sample image in RSSCN7 database and its corresponding three view features.

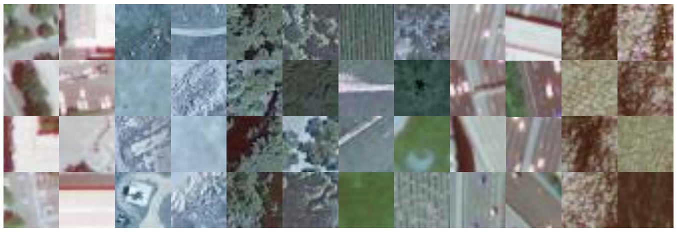


Fig. 6. Several samples of SAT-6 database. From left to right, every two columns belong to one class.

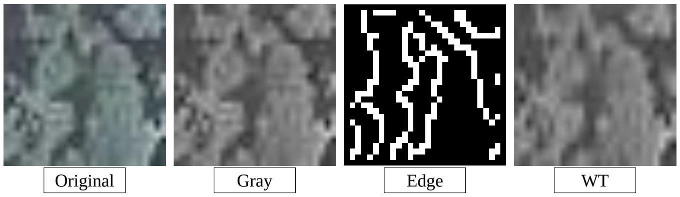


Fig. 7. One sample image in SAT-6 database and its corresponding gray, edge and WT features.

images (3000 images for each class) and these entries are randomly selected from the training set of SAT-6 database. Several examples in our subset are shown in Figure 6, and one selected image and its three-view features (a gray sub-image, an edge sub-image and a low frequency WT sub-image) are elucidated in Figure 7.

4.2 Experimental Results

In this section, we compare the proposed TCCANet and MS-TCCANet with several highly related works covering PCANet, CCANet, MS-CCANet and MCCANet for multi-view RS scene recognition. These algorithms contain several common parameters like the number of training samples, the filter number, the filter size and the block size. For fair comparison, the default parameter settings are identical for all these algorithms referring to Table 1 for precise values. In the following paragraphs, the impact of different parameters is discussed in the cross validation manner. That means, when we check a certain parameter, all other parameters are fixed to their defaults. Additionally, since we have three-view features, i.e., gray sub-images, edge sub-images and WT sub-images, they also form three combinations for single-view and two-view methods including PCANet, CCANet and MS-CCANet. To make the experiments more convincing, all possible combinations are tested in the following parts. We evaluate the RS scene classification performance by the recognition accuracy, which is the ratio of the number of correctly predicted test samples $Test_{correct}$ to the total number of test samples $Test_{total}$.

$$Recognition\ Accuracy = \frac{Test_{correct}}{Test_{total}} \times 100\% \qquad (33)$$

TABLE 1
The default parameter settings

Parameters	RSSCN7	SAT-6
♯ of Training samples	$N = 1000$	$N = 8000$
Filter Number	$L_1 = L_2 = 8$	$L_1 = L_2 = 8$
Filter Size	$k_1 \times k_2 = 5 \times 5$	$k_1 \times k_2 = 5 \times 5$
Block Size	$b_1 \times b_2 = 31 \times 31$	$b_1 \times b_2 = 27 \times 27$

4.2.1 The number of training samples

In this part, we investigate the influence of the size of the training set on the recognition results. For the RSSCN7 database, $N = \{1000, 2000\}$ images are randomly selected as the training set and the rest constitutes the testing set. For the SAT-6 database, the number of training samples varies from 8000 to 16000. For each parameter tunning, we conduct the experiment for 10 times and select the training samples by random permutations in each time. Table 2 lists the mean recognition rate and standard deviation of ten runs for PCANet, CCANet, MS-CCANet, MCCANet, TCCANet and MS-TCCANet on RSSCN7 and SAT-6 database. The best results are marked in bold. Table 2 illustrates that: (1) with the increase of N, the recognition rate generally goes up for all algorithms and different feature combinations; (2) Our MS-TCCANet achieves the highest recognition rate for all changes of the training set. Specifically, our MS-TCCANet outperforms the average performance of the famous PCANet by a large margin on both RSSCN7 (11.41%) and SAT-6 (19.53%) under default parameter settings (Table 1); (3) TCCANet also makes a significant improvement compared with all previous works including PCANet, CCANet, MS-CCANet and MCCANet; (4) Our previous work, i.e., MS-CCANet, also achieves prominent performance in SAT-6 database, suggesting the effectiveness of the multiple scale feature; (5) PCANet performs much poor when the 'edge' feature is utilized. Therefore, the 'PCANet (edge)' case is abandoned in the following experiments. This may imply that the 'edge' feature is difficult to distinguish in scene classification tasks, so other refined feature may yield a higher recognition rate. Nevertheless, it is still reasonable to use the 'edge' feature in our multiview experiments, since it provides a specific view of complementary information.

TABLE 2
RECOGNITION RESULTS WITH DIFFERENT NUMBERS OF TRAIN SET ON RSSCN7 AND SAT-6 DATASET

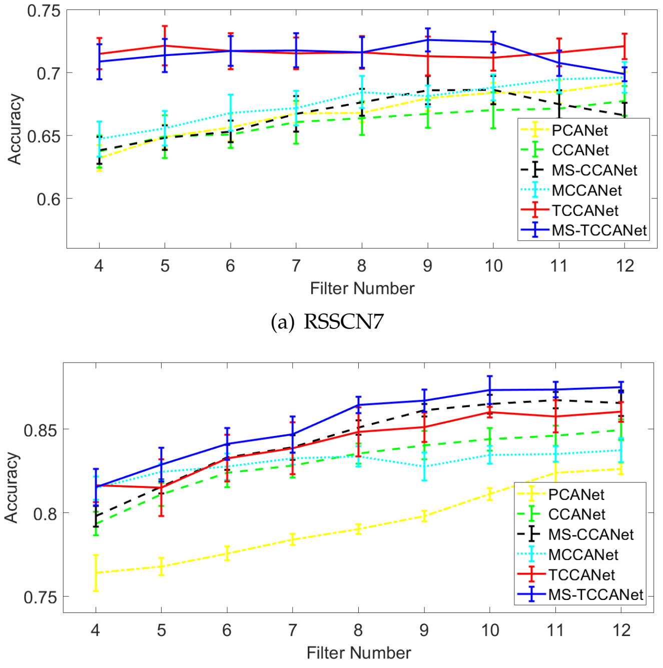
Dataset		RSSCN7		SAT-6	
# of training samples N		1000	2000	8000	16000
PCANet	gray	67.21% ± 0.79%	69.86% ± 1.50%	79.26% ± 0.26%	81.35% ± 1.41%
	WT	67.52% ± 0.85%	70.51% ± 1.75%	78.86% ± 0.44%	80.94% ± 0.95%
	edge	47.13% ± 0.73%	50.64% ± 1.43%	41.12% ± 0.96%	41.99% ± 1.90%
CCANet	WT & edge	65.99% ± 1.58%	70.70% ± 1.70%	82.51% ± 1.16%	83.49% ± 0.70%
	edge & gray	66.81% ± 1.13%	69.83% ± 2.00%	84.14% ± 0.71%	84.89% ± 1.15%
	WT & gray	66.31% ± 1.28%	68.31% ± 2.16%	83.79% ± 0.50%	84.27% ± 0.98%
MS-CCANet	WT & edge	66.83% ± 1.31%	70.86% ± 1.54%	83.80% ± 0.97%	84.95% ± 0.90%
	edge & gray	67.23% ± 1.04%	71.28% ± 1.78%	85.18% ± 0.81%	86.64% ± 1.03%
	WT & gray	69.48% ± 0.77%	71.39% ± 1.51%	85.85% ± 0.48%	86.41% ± 1.07%
MCCANet	gray & edge & WT	68.43% ± 1.28%	71.43% ± 1.38%	83.47% ± 0.54%	84.67% ± 0.78%
TCCANet	gray & edge & WT	71.80% ± 0.99%	73.09% ± 1.83%	85.23% ± 0.52%	86.18% ± 1.05%
MS-TCCANet	gray & edge & WT	**72.03% ± 0.71%**	**74.23% ± 1.63%**	**85.94% ± 0.80%**	**87.03% ± 2.26%**

[1] A±B: A is the ten-run mean recognition accuracy and B denotes the standard deviation.

4.2.2 The number of filters

In each convolutional layer, the number of filters can be adjusted. In this part, we change the number of filters of first layer L_1 from 4 to 12 and fixed the second layer $L_2 = 8$, and then tune L_2 by fixing $L_1 = 8$. All other parameters are kept to their default settings. Particularly, the numbers of training images are restored to 1000 and 8000 for RSSCN7 and SAT-6 database, respectively. For each parameter setting, we repeat the experiments for 10 times and divide the training set and testing set randomly in each time. For PCANet, CCANet and MS-CCANet, we report their average recognition results of different feature combinations to make the result easy reading.

Figure 8 shows the experimental results with different values of L_1. Figure 8(a) gives the results on RSSCN7 database. From Figure 8(a), we can see that both TCCANet and MS-TCCANet acquire remarkable improvement compared with all other counterparts. Generally speaking, TCCANet and MS-TCCANet are more robust to the variation of filter numbers than the previous works on the RSSCN7 dataset. Figure 8(b) reveals the results on SAT-6 database. It can be observed from Figure 8(b) that all methods carry out a better performance with the increase of L_1, owing to the fact that more filters produce a more comprehensive local feature. Besides, MS-TCCANet attains the top result all along. A notable difference between Figure 8(a) and Figure 8(b) is that the proposed algorithms achieve high accuracy even with small number of filters on the RSSCN7 dataset. The reason is that the image size of RSSCN7 (64 × 64) is larger than SAT-6 (28 × 28), which encodes much more pixel local information into the sample matrices. After decomposing the high-order covariance tensor, the first few filters can capture enough discriminant information for classification. Figure 9 exhibits the impact of L_2 on RSSCN7 dataset in Figure 9(a) and SAT-6 dataset in Figure 9(b). As shown in Figure 9(a), the proposed TCCANet and MS-TCCANet outperform other comparison algorithms in most scenarios. From Figure 9(b), we can see that both MS-TCCANet and MS-CCANet accomplish a brilliant outcome, which confirms the efficacy of multi-scale feature once again.



(a) RSSCN7

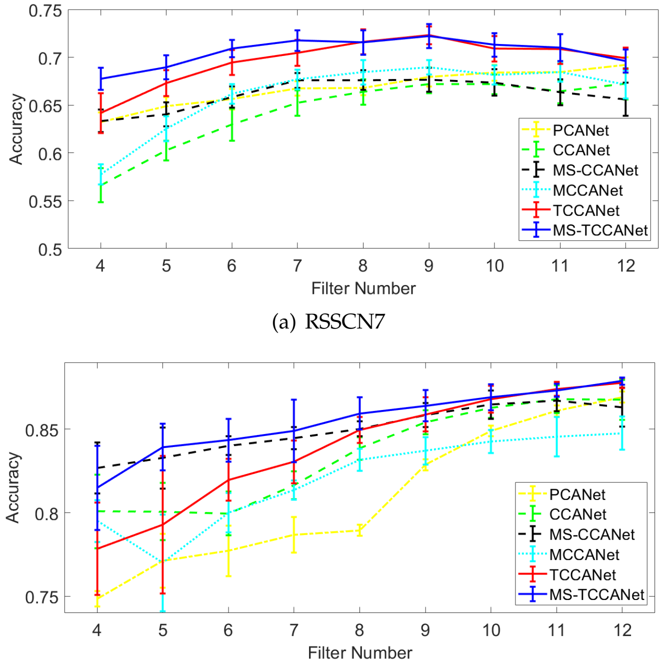
(b) SAT-6

Fig. 8. The experimental result of MS-TCCANet, TCCANet, MCCANet, MS-CCANet, CCANet and PCANet on (a) RSSCN7 database and (b) SAT-6 database under different number of filters L_1 with fixed $L_2 = 8$.

4.2.3 The filter size

In this part, we examine the effectiveness of different feature learning models by changing the filter size. The filter size of each convolutional layer actually depends on the patch size when we construct multi-view sample matrices $k_1 \times k_2$. In this experiment, the filter size is tunned in the range of $\{3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9, 11 \times 11\}$. For each change of filter size, all the algorithms are tested for 10 times. The ten-run average classification accuracy and the standard deviation are illustrated in Figure 10. Precisely, the Figure 10(a) and Figure 10(b) shows the results on RSSCN7 database and SAT-6 database, respectively. In each subfigure, the x-axis represents the filter size and y-axis represents the classification accuracy.

From Figure 10, we can see that with the increase of filter size, the recognition accuracy of all the feature learning models tend to decline. The reason is that large filters ignore the important local features for RS images with small size (28 × 28 or 64 × 64). With the increase of image sizes (e.g.,



(a) RSSCN7

(b) SAT-6

Fig. 9. The experimental results of MS-TCCANet, TCCANet, MCCANet, MS-CCANet, CCANet and PCANet on (a) RSSCN7 database and (b) SAT-6 database under different numbers of filters L_2 with fixed $L_1 = 8$.

$10,000 \times 10,000$) or the decrease of image Ground Sampling Distance (GSD), larger filters may gradually take the advantage. Additionally, Figure 10(a) suggests that the proposed MS-TCCANet is consistently superior to the comparison methods for all filter sizes. Figure 10(b) demonstrates that the MS-TCCANet achieves the highest accuracy on SAT-6 dataset at most cases except when the filter size is 3×3. However, the differences among top-performing methods are very trivial on the filter size of 3×3, which then become more significant after the filter size increases.

4.2.4 The block size

In the output layer, the final feature is learned based on the block-wise histogram strategy. In this part, we explore the influence of the block size. Specifically, the block size is changed from 3×3 to 35×35 for the RSSCN7 database and from 3×3 to 27×27 for the SAT-6 database. Similarly, with each parameter change, we run the experiment for 10 times. The mean recognition results and the corresponding standard deviations are given in Figure 11.

Figure 11(a) reveals the influence of block size on RSSCN7 database. From Figure 11(a), we can see that the recognition rates of all the algorithms are enhanced with the increase of block size. The proposed TCCANet and MS-TCCANet have prominent performance when $b_1 \times b_2 \geq 19 \times 19$. This superiority may not be as conspicuous as it does due to the range of y-axis.

Figure 11(b) shows the experimental results on SAT-6 database with different block sizes. From Figure 11(b), we can see that the MS-CCANet with the combination of gray and WT feature, achieves a outstanding result when $b_1 \times b_2 < 19 \times 19$. After that, our MS-TCCANet gradually gains advantage. Besides, the performance of all the algorithms is even poor when $b_1 \times b_2 = 19 \times 19$. The reason may be that this block size with the overlapping ratio (0.5) exactly ignores some important local information, e.g., the area around the center of an image, by considering that the image size of SAT-6 database is 28×28.

TABLE 3
RUNNING TIME (SECONDS) COMPARISON

Methods	Training time		Average testing time	
	RSSCN7	SAT-6	RSSCN7	SAT-6
PCANet	106.01	693.84	0.17	0.16
CCANet	39.75	125.43	0.03	0.01
MS-CCANet	35.28	140.62	0.03	0.01
MCCANet	77.90	142.89	0.05	0.01
TCCANet	127.20	172.77	0.06	0.01
MS-TCCANet	150.83	175.82	0.09	0.01

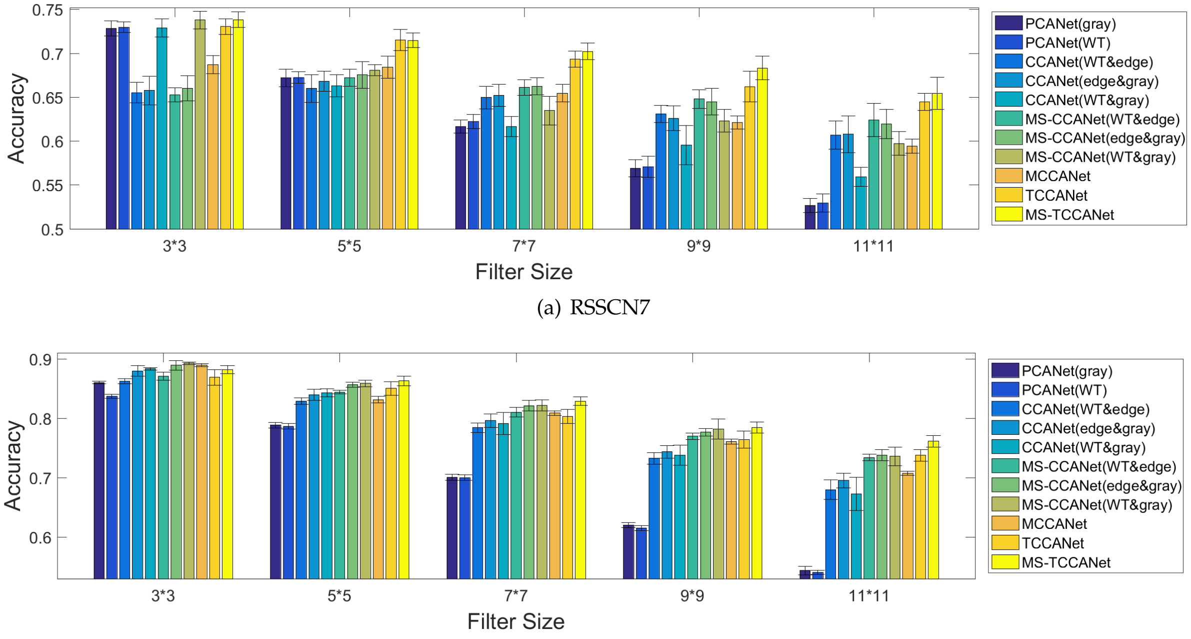
4.2.5 Running time

We report the code running time of different methods in Table 3 under the default parameter settings (see Table 1). All methods are implemented on Red Hat Enterprise Linux Workstatioin 7.7 (Maipo) with 2.7GHz CPU frequency and 176GB memory. Table 3 illustrates that the proposed TCCANet and MS-TCCANet consume more training time than low-order multi-view methods, i.e., CCANet, MS-CCANet and MCCANet on both RSSCN7 and SAT-6 datasets. This is because the high-order-correlation construction and tensor decomposition are computationally expensive than matrix operations. It is worth mentioning that we employ the CPU parallel computation to accelerate the TCCANet and MS-TCCANet in calculating the covariance tensor, i.e., Eq. (10). In comparison with the single view PCANet, our TCCANet needs comparable or even less running time in terms of both training and testing. In addition, the TCCANet and MS-TCCANet show a slight time cost growth towards the increase of training samples from 1000 (RSSCN7) to 8000 (SAT-6), comparing with all the previous methods. This may indicate that our TCCANet and MS-TCCANet are more feasible for large-scale RS recognition problems.

4.3 Comparison with deep CNN methods

In this section, we compare our TCCANet and MS-TCCANet with several deep CNN models, including VGGNet (VGG16 and VGG19) [51], ResNet (ResNet50 and ResNet101) [61] and SENet (SE-DenseNet) [62]. We use ADAM [63] as the parameter optimizer and implement all these networks using Keras[1]. The ADAM learning rate is set to 10^{-3}, and all the rest parameters are set as default. For fair comparison, we randomly select 1000 and 8000 training samples from RSSCN7 and SAT-6 datasets, respectively. The rest images are used for testing. We resize all images into 224×224 before feeding them into CNNs models. During the network training, we fix the "patience" parameter as 10. This means we will stop the training phrase if the recognition accuracy is not improved for ten consecutive epochs. Table 4 shows the classification results on gray and WT features of the two RS scene datasets.

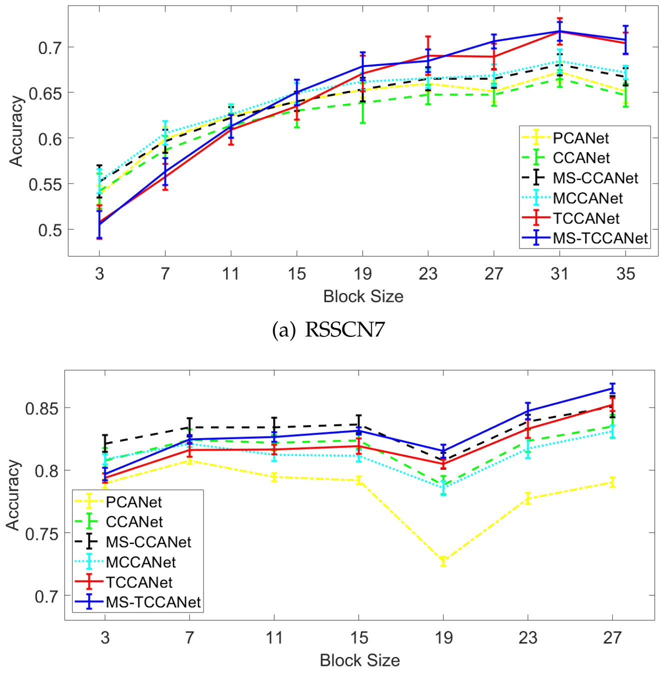
[1] https://keras.io/



(a) RSSCN7

(b) SAT-6

Fig. 10. The experimental results of MS-TCCANet, TCCANet, MCCANet, MS-CCANet, CCANet and PCANet on (a) RSSCN7 database and (b) SAT-6 database under different filter sizes.



(a) RSSCN7

(b) SAT-6

Fig. 11. The experimental result of MS-TCCANet, TCCANet, MCCANet, MS-CCANet, CCANet and PCANet on (a) RSSCN7 database and (b) SAT-6 database under different block size.

TABLE 4
COMPARISON WITH DEEP CNN METHODS (OUR TCCANET AND MS-TCCANET USE GRAY, WT AND EDGE FEATURES). OUR METHODS CAN ACHIEVE BETTER OR ON PAR CLASSIFICATION RESULTS COMPARED WITH MOST DEEP CNN MODELS BY USING MUCH LESS TRAINING TIME.

Methods	RSSCN7		SAT-6	
	gray	WT	gray	WT
VGG16	53.91%	54.69%	85.55%	91.41%
VGG19	55.86%	73.83%	83.59%	87.89%
ResNet50	56.25%	67.97%	79.69%	86.33%
ResNet101	57.42%	63.67%	83.98%	81.64%
SE-DenseNet121	71.88%	71.48%	82.81%	89.06%
TCCANet	71.80%		85.23%	
MS-TCCANet	72.03%		85.94%	

From Table 4 we can see that our MS-TCCANet achieves the highest recognition accuracy on both RSSCN7 and SAT-6 datasets when the gray feature is used. For the WT feature, our TCCANet and MS-TCCANet can also perform on par with most of the very deep CNN methods. The important thing to note is that the network structure of both TCCANet and MS-TCCANet are super lightweight, i.e., with only two convolutional layers. A natural benefit is that the training time is greatly reduced compared with VGGNet, ResNet and SENet. For example, it takes more than a day and a half to train ResNet101 on SAT-6 gray (42.54 hours) and WT (44.85 hours). While training our TCCANet on SAT-6 needs only 172.77 seconds as shown in Table 3. The efficient network training and high classification performance are two main strengths of our TCCANet and MS-TCCANet for RS images scene recognition than those popular CNN models with very deep structures.

5 CONCLUSION

Convolutional neural network (CNN) has been proven a successful model for hierarchical feature learning in both theoretical principle and real world applications. In recent years, many variants of CNN were proposed, such as PCANet, CCANet, MS-CCANet and MCCANet. The

PCANet seeks the filter bank by maximizing the sample separation and achieves great success in image classification. However, PCANet is only suitable for the single view data, so it is limited when the RS information stemming from diverse sources. Despite the fact that CCANet, MS-CCANet and MCCANet can be applied to two or more view cases, they take only pair-wise correlations into consideration and ignores the high-order-correlations among multi-view data. In this paper, we have presented tensor canonical correlation analysis networks (TCCANet) to solve this problem for multi-view RS scene recognition. Particularly, TCCANet discovers the high-order-correlation by directly maximizing all views' canonical correlation and seeks the optimal filter banks by analyzing a covariance tensor. In the output layer, the binarization and histogram are introduced as the non-linear processing and feature pooling. Furthermore, we also put forward a multiple scale development of TCCANet, i.e., MS-TCCANet. Finally, we carefully conducted the experiments on two real world datasets: RSSCN7 and SAT-6 data. Extensive experimental results showed that the proposed TCCANet and MS-TCCANet are statistically superior to PCANet, CCANet, MS-CCANet and MCCANet for multi-view RS scene recognition.

In the future, we will research on more efficient tensor decomposition methods to accelerate the ALS and reduce the TCCANet time consuming. In addition, validating our algorithms on RS images with multiple view angles and scales are also promising work directions as they are the two main variables when satellite passes. This future work shall be performed on datasets that not only organize data by their class label but also by the multi-view properties, such as view angle and scale features.

Appendix A
Proof of Theorem 1

Proof. By exploring the element-wise product, we have the following decomposition for the left side.

$$\begin{aligned}\rho_{12\cdots V} &= (z_1 \odot z_2 \odot \cdots \odot z_V)^T \mathbf{e} = \sum_{i=1}^{Nwh} z_1(i) z_2(i) \cdots z_V(i) \\ &= \sum_{i=1}^{Nwh} \prod_{v=1}^{V} z_v(i) = \sum_{i=1}^{Nwh} \prod_{v=1}^{V} \left(\sum_{j_v=1}^{k_1 k_2} \mathbf{x}_i^v(j_v) \alpha_v(j_v) \right),\end{aligned} \tag{34}$$

where $z_v(i)$ represents the i^{th} element of z_v, and this representation is also applied to $\mathbf{x}_i^v$ and α_v. According to the definition of tensor (outer) product, we have

$$\begin{aligned}\mathcal{C}_{12\cdots V}(j_1, j_2, \cdots, j_V) &= \sum_{i=1}^{Nwh} \mathbf{x}_i^1(j_1)\mathbf{x}_i^2(j_2) \cdots \mathbf{x}_i^V(j_V) \\ &= \sum_{i=1}^{Nwh} \prod_{v=1}^{V} \mathbf{x}_i^v(j_v)\end{aligned} \tag{35}$$

Additionally, according to the v-mode tensor-vector product defined in (6), we have

$$\begin{aligned}&(\mathcal{C}_{12\cdots V} \bar{\times}_v \alpha_v^T)(j_1, \cdots, j_{v-1}, j_{v+1}, \cdots, j_V) \\ &= \sum_{j_v=1}^{k_1 k_2} \mathcal{C}_{12\cdots V}(j_1, j_2, \cdots, j_V) \alpha(j_v) \\ &= \sum_{i=1}^{Nwh} \sum_{j_v=1}^{k_1 k_2} \left(\prod_{v=1}^{V} \mathbf{x}_i^v(j_v) \right) \alpha(j_v)\end{aligned} \tag{36}$$

Accordingly, the right-hand side of (12) equals to

$$\begin{aligned}&\mathcal{C}_{12\cdots V} \bar{\times}_1 \alpha_1^T \bar{\times}_2 \alpha_2^T \cdots \bar{\times}_V \alpha_V^T \\ &= \sum_{i=1}^{Nwh} \prod_{v=1}^{V} \left(\sum_{j_v=1}^{k_1 k_2} \mathbf{x}_i^v(j_v) \alpha(j_v) \right)\end{aligned} \tag{37}$$

Proof completed. □

Appendix B
Proof of problem (13) and (14) are equivalent

Proof. It is obvious that the constraints are equivalent. We then proof the objectives of problem (13) and (14) are equivalent.

$$\begin{aligned}&\mathcal{C}_{12\cdots V} \bar{\times}_1 \alpha_1^T \bar{\times}_2 \alpha_2^T \cdots \bar{\times}_V \alpha_V^T \\ &= \alpha_V^T \mathcal{C}_{(V)} (\alpha_{V-1} \otimes \cdots \otimes \alpha_2 \otimes \alpha_1) \\ &= h_V^T \widetilde{S}_{VV}^{-\frac{1}{2}} \mathcal{C}_{(V)} \Big((\widetilde{S}_{V-1,V-1}^{-\frac{1}{2}} h_{V-1}) \otimes \cdots \otimes (\widetilde{S}_{1,1}^{-\frac{1}{2}} h_1) \Big) \\ &= h_V^T \Big(\widetilde{S}_{VV}^{-\frac{1}{2}} \mathcal{C}_{(V)} (\widetilde{S}_{V-1,V-1}^{-\frac{1}{2}} \otimes \cdots \otimes \widetilde{S}_{1,1}^{-\frac{1}{2}}) (h_{V-1} \otimes \cdots \otimes h_1) \Big) \\ &= h_V^T \mathcal{K} (h_{V-1} \otimes \cdots \otimes h_2 \otimes h_1) \\ &= \mathcal{K} \bar{\times}_1 h_1^T \bar{\times}_2 h_2^T \cdots \bar{\times}_V h_V^T\end{aligned}$$

This accomplishes the proof. Some properties, such as, sequence of tensor-matrices product in (8) and Kronecker product in (9) are used in this proof. □

Acknowledgments

The authors would like to thank Professor Dacheng Tao for his insightful comments and discussions on this work.

References

[1] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[2] L. F. Osborne, "Continual crop development profiling using dynamical extended range weather forecasting with routine remotely-sensed validation imagery," Sep. 15 2015, uS Patent 9,131,644.

[3] A. M. Wilson and W. Jetz, "Remotely sensed high-resolution global cloud dynamics for predicting ecosystem and biodiversity distributions," *PLoS biology*, vol. 14, no. 3, p. e1002415, 2016.

[4] G. Melillos, K. Themistocleous, G. Papadavid, A. Agapiou, M. Prodromou, S. Michaelides, and D. G. Hadjimitsis, "Integrated use of field spectroscopy and satellite remote sensing for defence and security applications in cyprus," in *Fourth International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2016)*, vol. 9688. International Society for Optics and Photonics, 2016, p. 96880F.

[5] R. J. Grasso, "Defence and security applications of quantum cascade lasers," in *Optical Sensing, Imaging, and Photon Counting: Nanostructured Devices and Applications 2016*, vol. 9933. International Society for Optics and Photonics, 2016, p. 99330F.

[6] A. W. Kraft, B. A. Babenko, M. A. Baxter, and S. J. Bickerton, "Moving vehicle detection and analysis using low resolution remote sensing imagery," Apr. 9 2019, uS Patent 10,255,523.

[7] J. Saxena, R. Yadav, and S. Singh, "Remote traffic monitoring & control system," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2018, pp. 552–557.

[8] X. Wang, R. Song, A. Zhang, X. Ai, and J. Tao, "Remote sensing image magnification study based on the adaptive mixture diffusion model," *Information Sciences*, 2018.

[9] C. G. Homer, C. L. Aldridge, D. K. Meyer, and S. J. Schell, "Multiscale remote sensing sagebrush characterization with regression trees over wyoming, usa: laying a foundation for monitoring," *International Journal of Applied Earth Observation and Geoinformation*, vol. 14, no. 1, pp. 233–244, 2012.

[10] G. P. Asner, "Contributions of multi-view angle remote sensing to land-surface and biogeochemical research," *Remote Sensing Reviews*, vol. 18, no. 2-4, pp. 137–162, 2000.

[11] J. Rhee, J. Im, and G. J. Carbone, "Monitoring agricultural drought for arid and humid regions using multi-sensor remote sensing data," *Remote Sensing of Environment*, vol. 114, no. 12, pp. 2875–2887, 2010.

[12] X. Yang, W. Liu, D. Tao, J. Cheng, and S. Li, "Multiview canonical correlation analysis networks for remote sensing image recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1855–1859, Oct 2017.

[13] K. Kallio, T. Kutser, T. Hannonen, S. Koponen, J. Pulliainen, J. Veps?l?inen, and T. Pyh?lahti, "Retrieval of water quality from airborne imaging spectrometry of various lake types in different seasons," *Science of The Total Environment*, vol. 268, no. 1, pp. 59 – 77, 2001, lake water monitoring in. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048969700006859

[14] F. Pacifici, N. Longbotham, and W. J. Emery, "The importance of physical quantities for the analysis of multitemporal and multiangular optical very high spatial resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6241–6256, 2014.

[15] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, 2015.

[16] F. P. Luus, B. P. Salmon, F. Van den Bergh, and B. T. J. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2448–2452, 2015.

[17] N. He, M. E. Paoletti, J. M. Haut, L. Fang, S. Li, A. Plaza, and J. Plaza, "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 755–769, 2018.

[18] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[19] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 937–949, 2017.

[20] G. Li, L. Li, H. Zhu, X. Liu, and L. Jiao, "Adaptive multiscale deep fusion residual network for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8506–8521, 2019.

[21] Y. Zhang, Y. Lu, D. Zhang, L. Shang, and D. Wang, "Risksens: A multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1544–1553.

[22] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 46–58, 2020.

[23] X. Yang, W. Liu, D. Tao, and J. Cheng, "Canonical correlation analysis networks for two-view image recognition," *Information Sciences*, vol. 385, no. Supplement C, pp. 338 – 352, 2017.

[24] X. Yang and W. Liu, "Multiple scale canonical correlation analysis networks for two-view object recognition," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 325–334.

[25] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen, "Tensor canonical correlation analysis for multi-view dimension reduction," *IEEE transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3111–3124, 2015.

[26] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, Nov 2015.

[27] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: a learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 37.

[28] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, "Remote sensing image scene classification using bag of convolutional features," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1735–1739, 2017.

[29] J. A. dos Santos, O. A. B. Penatti, and R. da Silva Torres, "Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification." in *VISAPP (2)*, 2010, pp. 203–208.

[30] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3681–3693, July 2015.

[31] Y. Ke, R. Sukthankar *et al.*, "Pca-sift: A more distinctive representation for local image descriptors," *CVPR (2)*, vol. 4, pp. 506–513, 2004.

[32] W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang, "Object detection in high-resolution remote sensing images using rotation invariant parts based model," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 74–78, 2013.

[33] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*. IEEE, 2003, p. 1470.

[34] Q. Zhu, Y. Zhong, B. Zhao, G. Xia, and L. Zhang, "Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 6, pp. 747–751, June 2016.

[35] 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," *International Journal of Remote Sensing*, vol. 34, no. 1, pp. 45–59, 2013.

[36] L. Zhao, P. Tang, and L. Huo, "Feature significance-based multibag-of-visual-words model for remote sensing image scene classification," *Journal of Applied Remote Sensing*, vol. 10, no. 3, p. 035004, 2016.

[37] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1359–1371, July 2014.

[38] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[39] C. Rodarmel and J. Shan, "Principal component analysis for hyperspectral image classification," *Surveying and Land Information Science*, vol. 62, no. 2, pp. 115–122, 2002.

[40] Z. Lv, Y. Hu, H. Zhong, J. Wu, B. Li, and H. Zhao, "Parallel k-means clustering of remote sensing images based on mapreduce," in *International Conference on Web Information Systems and Mining*. Springer, 2010, pp. 162–170.

[41] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li, "Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 1, pp. 109–113, Jan 2012.

[42] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3660–3671, June 2016.

[43] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1017–1027, April 2017.

[44] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, Dec 2015.

[45] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[46] K. Nogueira, O. A. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539 – 556, 2017.

[47] K. Nogueira, W. O. Miranda, and J. A. Dos Santos, “Improving spatial feature representation from aerial scenes by using convolutional networks,” in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2015, pp. 289–296.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[49] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[51] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[52] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.

[53] X. Lu, H. Sun, and X. Zheng, “A feature aggregation convolutional neural network for remote sensing scene classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7894–7906, 2019.

[54] Q. Wang, S. Liu, J. Chanussot, and X. Li, “Scene classification with recurrent attention of VHR remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1155–1167, 2019.

[55] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition,” *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, Apr 2002.

[56] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, Dec 2006.

[57] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, Aug 2013.

[58] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, “On the best rank-1 and rank-(r1, r2, ..., rn) approximation of higher-order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[59] P. Comon, X. Luciani, and A. L. F. de Almeida, “Tensor decompositions, alternating least squares and other tales,” *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 393–405, 2009.

[60] J. D. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, Sep 1970.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[62] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” 2018.

[63] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

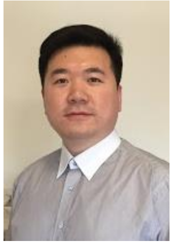
Xinghao Yang received the B.Eng. degree in electronic information engineering and M.Eng. degree in information and communication engineering from the China University of Petroleum (East China), Qingdao, China, in 2015 and 2018, respectively. Currently, he is a PhD student in Advanced Analytics Institute, University of Technology Sydney, Australia.

His research interests include multi-view learning and adversarial machine learning with publications on information fusion and information sciences.

Weifeng Liu (M’12, SM’17) received the double B.S. degrees in automation and business administration and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2002 and 2007, respectively.

He was a Visiting Scholar with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia, from 2011 to 2012. He is currently a Full Professor with the College of Information and Control Engineering, China University of Petroleum, Qingdao, China. He has authored or coauthored a dozen papers in top journals and prestigious conferences, including four Essential Science Indicators (ESI) highly cited papers and two ESI hot papers. His research interests include computer vision, pattern recognition, and machine learning.

Prof. Liu serves as an Associate Editor for the Neural Processing Letters, the Co-Chair for the IEEE SMC Technical Committee on Cognitive Computing, and a Guest Editor for special issue of the Signal Processing, theIET Computer Vision, the Neurocomputing, and the Remote Sensing. He also serves over 20 journals and over 40 conferences.



Wei Liu is the Data Science Program Leader and a Senior Lecturer at the School of Computer Science in the University of Technology Sydney (UTS), Australia. He obtained his PhD in machine learning research from the University of Sydney (USyd). Before joining UTS, he was a Research Fellow at the University of Melbourne (UniMelb). His research focuses on theoretical advancements of machine learning and data mining. He has published over 80 papers in top IEEE transactions and top conferences including KDD, AAAI, IJCAI, ICCV, ICDM, etc. He has 3 best paper awards.