IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 2, 2017, accepted February 2, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2675478

Multilinear Principal Component Analysis Network for Tensor Object Classification

JIASONG WU[1, 2], (Member, IEEE), SHIJIE QIU[1, 2], RUI ZENG[1, 2], (Member, IEEE), YOUYONG KONG[1, 2], LOTFI SENHADJI[2, 3, 4], (Senior Member, IEEE), AND HUAZHONG SHU[1, 2], (Senior Member, IEEE)

[1]LIST, Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, 210096 Nanjing, China
[2]Centre de Recherche en Information Biomedicale Sino-Francais, 210096 Nanjing, China
[3]INSERM, 35000 Rennes, France
[4]The Laboratoire Traitement du Signal et de l'Image, Universite de Rennes 1, 35000 Rennes, France

Corresponding author: J. Wu (jswu@seu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61201344, Grant 61271312, Grant 61401085, Grant 31571001, Grant 31640028, Grant 61572258, and Grant 11301074, in part by the Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, by the Qing Lan Project and the 333 Project under Grant BRA2015288, and in part by the Open Fund of China-USA Computer Science Research Center under Grant KJR16026.

ABSTRACT The recently proposed principal component analysis network (PCANet) has performed well with respect to the classification of 2-D images. However, feature extraction may perform less well when dealing with multi-dimensional images, since the spatial relationships within the structures of the images are not fully utilized. In this paper, we develop a multilinear principal component analysis network (MPCANet), which is a tensor extension of PCANet, to extract the high-level semantic features from multi-dimensional images. The extracted features largely minimize the intraclass invariance of tensor objects by making efficient use of spatial relationships within multi-dimensional images. The proposed MPCANet outperforms traditional methods on a benchmark composed of three data sets, including the UCF sports action database, the UCF11 database, and a medical image database. It is shown that even a simple one-layer MPCANet may outperform a two-layer PCANet.

INDEX TERMS Deep learning, MPCANet, PCANet, tensor object classification, medical image classification.

I. INTRODUCTION

A major difficulty of tensor object description in the visual world is the large amount of intra-class variability in tensor objects caused by illumination, rotation, scaling or more complex deformation. In recent years, learning multiple-level representations from visual content by convolutional neural networks (ConvNets) [1]–[6] has attracted much attention. Typically, ConvNets consist of multiple trainable stages stacked on top of each other followed by a supervised classifier. Each stage is composed of three layers: a convolution layer, a nonlinear processing layer, and a pooling layer.

While trainable ConvNets achieve success in many computer vision tasks, the computational complexity of backpropagation in the training phase and time-consuming fine tunes adversely affect efficiency of use. To address this issue, ConvNets without backpropagation has frequently been used by researchers. Recently, Chan *et al.* [7] proposed a new variant of convolutional networks, namely, principal component analysis network (PCANet), which does not make use of backpropagation to obtain features from visual content. The features of input visual content are extracted through three processing stages, i.e., principal component analysis (PCA) filter banks in convolution layers, binarization in nonlinear processing layers, and block-wise histogram in pooling layers, to avoid backpropagation. Although PCANet is constructed from the most basic units, it surprisingly achieves the state-of-the-art performance among most two-dimensional (2-D) image classification tasks. Due to the success of this non-propagation network, this sort of convolutional network has been extensively researched and many novelty works have been published recently. Feng *et al.* [8] proposed a discriminative locality alignment network (DLANet) for scene classification. Qin *et al.* [9] combined PCANet with spatial pyramid pooling (SP) for underwater live fish recognition. Zhao *et al.* [10] presented a multi-level modified finite radon transform network (MMFRTN) for image upsampling. Lei *et al.* [11] developed a stacked image descriptor for face recognition. Shi *et al.* [12] proposed a histopathological image classification method based on PCANet and matrix-form classifier. By combining linear regression classification model (LRC) and PCANet, Huang and Yuan [13] proposed a weighted-PCANet to solve the face recognition problem. Hao and Zhao [14] presented an incremental PCANet, which is a lifelong learning framework. Zheng *et al.* [15] investigated a deep learning technique, named deep learning with PCANet (DLPCANet), to estimate human age. Zeng *et al.* [16] proposed a quaternion principal component analysis network for color image classification. Wu *et al.* [17] proposed a Grassmann average network (GANet) algorithm to improve the robustness of learned features from images. They also proposed a quaternion GA network (QGANet) to fuse color information and learn superior representation for color histopathological images at the same time.

While these methods achieve success in their own fields, most methods represent multi-dimensional patches, taken from a tensor object, as one-dimensional (1-D) vectors to learn their features. However, 1-D representation may not only inevitably destroy the spatial structures of multidimensional objects; it may also suffer from the so-called curse of dimensionality [18]. Thus, a more compact representation, which can keep spatial information of structures in tensors, is necessary. In the last few years, some researchers [19]–[21] have shown that it is more suitable to represent multidimensional images as tensors rather than vectors. As a result, many research groups have shown great interest in the tensor extension of deep learning networks. For example, a tensor deep stacking network (TDSN), which is a tensor extension of traditional deep stacking network (DSN) proposed by Hutchinson *et al.* [22], has been applied to MNIST handwriting recognition and phone classification. Yu *et al.* [23] successfully extended deep neural network (DNN) to tensor space and obtained a deep tensor neural network (DTNN), which outperforms DNN in large vocabulary speech recognition. To the best of our knowledge, a similar tensor extension on PCANet has not been reported in the literature.

In this paper, we propose a multilinear principal component analysis network (MPCANet), a tensor extension of PCANet [7], for tensor object classification. For the proposed MPCANet, the convolution layers, the nonlinear processing layers, and the pooling layers consist of multilinear principal component analysis (MPCA) [19] filter banks, binarizations, and block-wise histograms, respectively. Proper experiments have been conducted on a UCF sports action database, a UCF11 database, and also a medical image database to evaluate the performance of MPCANet and other methods.

The rest of the paper is organized as follows. The MPCA algorithm is briefly introduced in Section II. Section III reviews the architecture of PCANet. Section IV describes the architecture of MPCANet in detail. The classification performance of MPCANet and PCANet is evaluated and compared on various tensor object datasets in Section V. Section VI concludes the paper.

II. REVIEW OF MPCA ALGORITHM

This section offers a brief review of the MPCA algorithm [19], which is a tensor extension of PCA, but it can capture more original tensor input variation than PCA.

An Nth-order tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, which is addressed by N indices $I_n, n = 1, 2, \ldots, N$, and each I_n addresses the n-mode of $\mathcal{X}$. The n-mode product of a tensor $\mathcal{X}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ is defined as:

$$(\mathcal{X} \times_n \mathbf{U})_{(i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N)} = \sum_{i_n} \mathcal{X}_{(i_1,\ldots,i_N)} \cdot \mathbf{U}_{(j_n,i_n)}. \tag{1}$$

By using the N projection matrices $\{\mathbf{V}^{(n)^T} \in \mathbb{R}^{P_n \times I_n}, n = 1, \ldots, N\}$, we can transform a set of M tensor objects $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}, m = 1, \ldots, M\}$ into another set of M tensor objects $\{\mathcal{Y}_m \in \mathbb{R}^{P_1 \times P_2 \times \cdots P_N}, m = 1, \ldots, M\}$, where $P_n < I_n, n = 1, 2, \ldots, N$. That is,

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{V}^{(1)^T} \times_2 \mathbf{V}^{(2)^T} \times \cdots \times_N \mathbf{V}^{(N)^T}, \quad m = 1, 2, \ldots, M. \tag{2}$$

Eq. (2) can also be expressed in matrix form as follows [19]:

$$\mathbf{X}_{m(n)} = \mathbf{V}^{(n)} \cdot \mathbf{Y}_{m(n)} \cdot \Big(\mathbf{V}^{(n+1)} \otimes \mathbf{V}^{(n+2)} \otimes \cdots \otimes \mathbf{V}^{(N)} \otimes \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes \cdots \otimes \mathbf{V}^{(n-1)}\Big)^T, \tag{3}$$

where $\otimes$ denotes the Kronecker product. $\mathbf{X}_{m(n)}$ is the unfolded matrix of the tensor $\mathcal{X}_m$ along the n-mode. That is to say, the column vectors of $\mathbf{X}_{m(n)}$ are the n-mode vectors of $\mathcal{X}_m$. $\mathbf{Y}_{m(n)}$ is the unfolded matrix of the tensor $\mathcal{Y}_m$ along the n-mode.

The objective of MPCA is the determination of N projection matrices $\{\mathbf{V}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \ldots, N\}$ that satisfy the following conditions:

$$\{\mathbf{V}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \ldots, N\} = \mathop{\arg\max}_{v^{(1)}, v^{(2)}, \ldots, v^{(N)}} \Psi_{\mathcal{Y}}, \tag{4}$$

where $\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \left\| \mathcal{Y}_m - \overline{\mathcal{Y}} \right\|_F^2$, $\overline{\mathcal{Y}}$ denotes the mean tensor calculated as $\overline{\mathcal{Y}} = (1/M) \sum_{m=1}^{M} \mathcal{Y}_m$. The problem of (4) has been solved by Lu *et al.* [19] as follows:

Let $\{\mathbf{V}^{(n)}, n = 1, \ldots, N\}$ be the solution of (4). Then, given all other projection matrices $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n-1)}, \mathbf{V}^{(n+1)}, \ldots, \mathbf{V}^{(N)}$, the matrix $\mathbf{V}^{(n)}$ consists of P_n eigenvectors corresponding to the largest P_n eigenvalues of the following matrix:

$$\mathbf{\Phi}^{(n)} = \sum_{m=1}^{M} \Big(\mathbf{X}_{m(n)} - \overline{\mathbf{X}}_{(n)}\Big) \mathbf{V}_{\mathbf{\Phi}^{(n)}} \mathbf{V}_{\mathbf{\Phi}^{(n)}}^T \Big(\mathbf{X}_{m(n)} - \overline{\mathbf{X}}_{(n)}\Big)^T, \tag{5}$$

where

$$\mathbf{V}_{\mathbf{\Phi}^{(n)}} = \Big(\mathbf{V}^{(n+1)} \otimes \mathbf{V}^{(n+2)} \otimes \cdots \otimes \mathbf{V}^{(N)} \; \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes \cdots \otimes \mathbf{V}^{(n-1)}\Big). \tag{6}$$

and $\mathbf{X}_{m(n)}$ is shown in (3), and $\overline{\mathbf{X}}_{(n)} = (1/M) \sum_{m=1}^{M} \mathbf{X}_{m(n)}$.

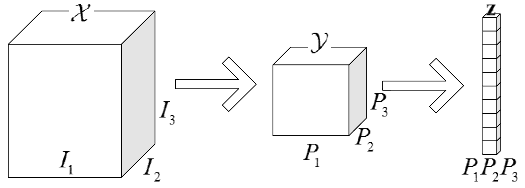


FIGURE 1. Visual illustration of MPCA flow diagram.

In practice, we generally further map the tensor $\mathcal{Y}_m \in \mathbb{R}^{P_1 \times P_2 \times \cdots \times P_N}$ to a vector $\mathbf{z}_m \in \mathbb{R}^{P_1 P_2 \cdots P_N}$, whose elements are arranged according to the amount of energy that they carry [19]. In practical implementation, the first L_1 elements that carry enough energy are chosen as features of the sample. Fig. 1 provides a simple visual illustration of MPCA algorithm for a third-order tensor. Note that the MPCA degrades to 2-D PCA [24] and PCA when dealing with the matrix (second-order tensor) and the vector (first-order tensor), respectively.

III. REVIEW OF PCANet

This section offers a brief review of the two-layer PCANet (PCANet-2), which is depicted in Fig. 2. The components of PCANet-2 is described in detail in the following.

A. THE FIRST STAGE OF PCANet-2

Assume that we have M input training images $\{\mathbf{I}_m \in \mathbb{R}^{I_1 \times I_2}, m = 1, \ldots, M\}$. The size of image patch (or 2D filter size) is set to $k_1 \times k_2$, where $1 < k_i < I_i$. By collecting all the (overlapping) $k_1 \times k_2$ image patches around each element of the mth image, we obtain $\mathbf{A}^1_m = \{\mathbf{A}^1_{m,q} \in \mathbb{R}^{k_1 \times k_2}, q = 1, \ldots, I_1 I_2\}$, where the superscript 1 denotes the 1st stage, and each $\mathbf{A}^1_{m,q}$ denotes the qth image patch of the original image $\mathbf{X}_m$. We reshape each image patch $\mathbf{A}^1_{m,q}$ into a vector $\mathbf{a}^1_{m,q} \in \mathbb{R}^{k_1 k_2}$. We then subtract patch mean from each image patch, and get $\overline{\mathbf{P}}^1_m = \left[\bar{\mathbf{a}}^1_{m,1}, \bar{\mathbf{a}}^1_{m,2}, \ldots, \bar{\mathbf{a}}^1_{m,I_1 I_2}\right]$, where $\bar{\mathbf{a}}^1_{m,q}$ is a mean-removed image patch. We construct a matrix to further collect these patches of all images, i.e.,

$$\mathbf{P}^1_m = [\overline{\mathbf{P}}^1_1, \overline{\mathbf{P}}^1_2, \ldots, \overline{\mathbf{P}}^1_M] \in \mathbb{R}^{k_1 k_2 \times M I_1 I_2}. \tag{7}$$

PCA is then utilized to minimize the reconstruction error as follows

$$\min_{\mathbf{V}^1 \in \mathbb{R}^{k_1 k_2 \times L_1}} \left\| \mathbf{P}^1_m - \mathbf{V}^1 \left(\mathbf{V}^1\right)^T \mathbf{P}^1_m \right\|^2_F, \\ \text{s.t. } \left(\mathbf{V}^1\right)^T \mathbf{V}^1 = \mathbf{E}_{L_1}, \tag{8}$$

where $\mathbf{E}_{L_1}$ is an unitary matrix of size $L_1 \times L_1$. The optimal solution $\tilde{\mathbf{V}}^1$ of (8) is known as the first L_1 principal eigenvectors of $\mathbf{P}^1_m \left(\mathbf{P}^1_m\right)^T$. Note that the eigenvalues are ordered as $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{L_1} \geq \ldots$. The PCA filters in the first stage can then be expressed as

$$\mathbf{W}^1_{l_1} = \mathrm{Mat}_{k_1,k_2}(\tilde{\mathbf{v}}^1_{l_1}) \in \mathbb{R}^{k_1 \times k_2}, l_1 = 1, 2, \ldots, L_1, \tag{9}$$

where $\tilde{\mathbf{v}}^1_{l_1}$ denotes the l_1-th principal eigenvector of $\tilde{\mathbf{V}}^1$, and $\mathrm{Mat}_{k_1,k_2}(\tilde{\mathbf{v}}^1_{l_1})$ is a function that maps a vector $\tilde{\mathbf{v}}^1_{l_1} \in \mathbb{R}^{k_1 k_2}$ to a matrix $\mathbf{W}^1_{l_1} \in \mathbb{R}^{k_1 \times k_2}$.

The l_1-th output feature map of the first stage is

$$\mathbf{F}^{l_1}_m = \mathbf{I}_m * \mathbf{W}^1_{l_1}, m = 1, 2, \ldots, M, \tag{10}$$

where $*$ denotes the two-dimensional (2D) convolution. Note that the feature matrix $\mathbf{F}^{l_1}_m$ has the same size as the original image $\mathbf{I}_m$ since the zero-padding method is used first for the original image $\mathbf{I}_m$ before performing the sliding process of image patches.

Therefore, for each input image $\mathbf{I}_m \in \mathbb{R}^{I_1 \times I_2}$, we obtain L_1 output images $\left\{\mathbf{F}^{l_1}_m \in \mathbb{R}^{I_1 \times I_2}, l_1 = 1, 2, \ldots, L_1\right\}$ after the first stage of PCANet.

B. THE SECOND STAGE OF PCANet-2

Similar to the first stage, we collect all the patches of $\mathbf{F}^{l_1}_m$ and then subtract patch mean from each patch, and for each $\mathbf{F}^{l_1}_m$ we can get $\overline{\mathbf{P}}^2_{m,l_1} = \left[\bar{\mathbf{a}}^2_{m,l_1,1}, \bar{\mathbf{a}}^2_{m,l_1,2}, \ldots, \bar{\mathbf{a}}^2_{m,l_1,I_1 I_2}\right] \in \mathbb{R}^{k_1 k_2 \times I_1 I_2}$, where the superscript 2 denotes the 2st stage. Then for all M input images, we obtain $\mathbf{P}^2_{m,l_1} = \left[\overline{\mathbf{P}}^2_{1,l_1}, \overline{\mathbf{P}}^2_{2,l_1}, \ldots, \overline{\mathbf{P}}^2_{M,l_1}\right] \in \mathbb{R}^{k_1 k_2 \times M I_1 I_2}$. We then concatenate $\mathbf{P}^2_{m,l_1}, l_1 = 1, 2, \ldots, L_1$ together and obtain $\mathbf{P}^2_m = \left[\mathbf{P}^2_{m,1}, \mathbf{P}^2_{m,2}, \ldots, \mathbf{P}^2_{m,L_1}\right] \in \mathbb{R}^{k_1 k_2 \times L_1 M I_1 I_2}$.

By repeating the same process as the first stage, we can get the PCA filters of the second stage as follows

$$\mathbf{W}^2_{l_2} = \mathrm{Mat}_{k_1,k_2}(\tilde{\mathbf{v}}^2_{l_2}) \in \mathbb{R}^{k_1 \times k_2}, \quad l_2 = 1, 2, \ldots, L_2, \tag{11}$$

where $\tilde{\mathbf{v}}^2_{l_2}$ denotes the $l_{\neq}$-th principal eigenvector of $\mathbf{P}^2_m \left(\mathbf{P}^2_m\right)^T$.

Then, the l_2-th output feature map of the second stage is

$$\mathbf{G}^{l_2}_{m,l_1} = \mathbf{F}^{l_1}_m * \mathbf{W}^2_{l_2}, \quad m = 1, 2, \ldots, M, \tag{12}$$

Therefore, for each input image $\mathbf{F}^{l_1}_m \in \mathbb{R}^{I_1 \times I_2}$, we obtain L_2 output images $\left\{\mathbf{G}^{l_2}_{m,l_1} \in \mathbb{R}^{I_1 \times I_2}, l_2 = 1, 2, \ldots, L_2\right\}$ after the second stage of PCANet.

C. THE OUTPUT STAGE OF PCANet-2

At the output stage, we first use the function $H(.)$, whose value is one for positive entries and zero otherwise, to binarize the results of the second PCA stage, and then convert the L_2 outputs in $\left\{\mathbf{G}^{l_2}_{m,l_1}, l_2 = 1, 2, \ldots, L_2\right\}$ back into a single integer-valued "image":

$$\mathbf{T}^{l_1}_m = \sum_{l_2=1}^{L_2} 2^{l_2-1} H\left(\mathbf{G}^{l_2}_{m,l_1}\right). \tag{13}$$

Note that each pixel value in "image" $\mathbf{T}^{l_1}_m$ is an integer in the range $[0, 2^{L_2-1}]$. We then use B blocks to split these feature maps and compute the histogram (with 2^{L_2} bins) of the decimal values in each block. We denote the result of this

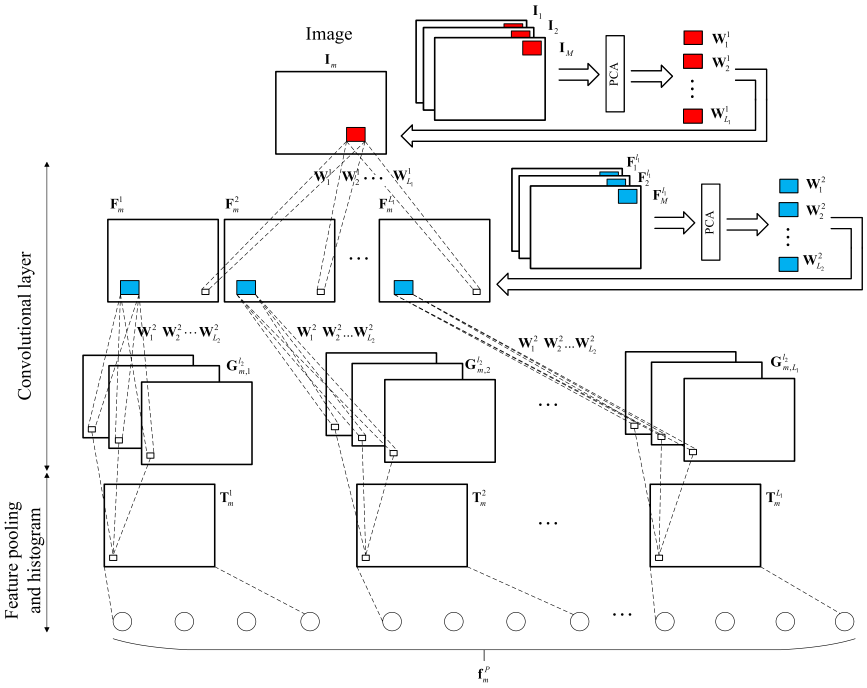


FIGURE 2. The architecture of two-layer PCANet (PCANet-2).

process as Bhist($\mathbf{T}_m^{l_1}$). We concatenate all the histograms into one vector and obtain the final feature of PCANet:

$$\mathbf{f}_m^P = [\text{Bhist}(T_m^1), \ldots, \text{Bhist}(T_m^{L_1})]^T \in \mathbb{R}^{(2^{L_2})L_1B}. \quad (14)$$

Note that the local block can be either overlapping or non-overlapping depending on application [7]. The feature vector is then sent to a support vector machine (SVM) classifier [25]–[27].

As depicted in [7], when assuming $I_1I_2 \geq \max(k_1, k_2, L_1, L_2, B)$, the overall complexity of the PCANet-2 is $O(I_1I_2k_1k_2(L_1+L_2)+I_1I_2(k_1k_2)^2)$.

IV. THE ARCHITECTURE OF THE PROPOSED MPCANet

The architecture of our proposed two-layer MPCANet (MPCANet-2) is summarized in Fig. 3. In this section, we describe each of its components in detail.

A. THE FIRST STAGE OF MPCANet-2

Assume that we have a set of M third-order tensor objects $\{\mathcal{X}_m \in \mathbb{R}^{I_1\times I_2\times I_3}, m = 1, \ldots, M\}$ for training. The size of tensor patch (or 3D filter size) is set to $k_1 \times k_2 \times k_3$, where $1 < k_i < I_i$. By collecting all the (overlapping) $k_1 \times k_2 \times k_3$ tensor patches around each element of the mth tensor object, we obtain $\mathcal{A}_m^1 = \{\mathcal{A}_{m,q}^1 \in \mathbb{R}^{k_1\times k_2\times k_3}, q = 1, \ldots, I_1I_2I_3\}$, where the superscript 1 denotes the 1st stage, and each $\mathcal{A}_{m,q}^1$ denotes the qth tensor patch of the original tensor $\mathcal{X}_m$. Note that a tensor patch is a 3D matrix extracted from a tensor object and can be seen as a 3D tensor extension of 2D image patch in PCANet. There are two reasons why we use a tensor patch: (1) tensor is more likely to represent abstract semantic than the ordinary 2D matrix in terms of high-dimensional data; (2) a tensor patch can capture some local information from the original tensor object.

By repeating the above procedure for all M tensor objects, we can get $\mathcal{A}^1 = \{\mathcal{A}_m^1, m = 1, \ldots, M\}$, which is then processed by MPCA algorithm [19], which is briefly described in Eqs. (3)-(6), to obtain three projection matrices $\mathbf{V}_1 = \{\mathbf{V}_1^{(n)} \in \mathbb{R}^{k_n\times p_n}, n = 1, 2, 3\}$. The projected tensor patch is therefore expressed as

$$\mathcal{S}_{m,q}^1 = \mathcal{A}_{m,q}^1 \times_1 \mathbf{V}_1^{(1)^T} \times_2 \mathbf{V}_1^{(2)^T} \times_3 \mathbf{V}_1^{(3)^T} \in \mathbb{R}^{p_1\times p_2\times p_3}, \quad (15)$$

which is then converted to a vector $\mathbf{z}_{m,q}^1 \in \mathbb{R}^{p_1p_2p_3}$. Suppose the number of filter banks in stage i is L_i; we then pick up the first L_i elements of $\mathbf{z}_{m,q}^1$ to form a new vector $\hat{\mathbf{z}}_{m,q}^1 \in \mathbb{R}^{L_1}$. Similarly, by constructing the matrices for all $I_1 \times I_2 \times I_3$ projected tensor patches and putting them together, we can obtain

$$\mathbf{z}_m^1 = \begin{bmatrix} \hat{\mathbf{z}}_{m,1}^1 & \hat{\mathbf{z}}_{m,2}^1 & \cdots & \hat{\mathbf{z}}_{m,I_1I_2I_3}^1 \end{bmatrix}^T \in \mathbb{R}^{I_1I_2I_3\times L_1}. \quad (16)$$

Then, the inverse of the previous mapping to the columns of $\mathbf{z}_m^1$ is used to derive a set of tensor objects $\left\{\mathcal{F}_m^{l_1} \in \mathbb{R}^{I_1\times I_2\times I_3}, l_1 = 1, \ldots, L_1\right\}$, where $\mathcal{F}_m^{l_1}$ is the l_1th tensor feature output of $\mathcal{X}_m$ in the first stage. That is, we reshape each column of $\mathbf{z}_m^1$ back to a three-dimensional (3D) tensor $\mathcal{F}_m^{l_1}$. Note that the tensor feature $\mathcal{F}_m^{l_1}$ has the same size as the original tensor $\mathcal{X}_m$ since the zero-padding method is used first for the original tensor $\mathcal{X}_m$ before performing the sliding process of tensor patches.

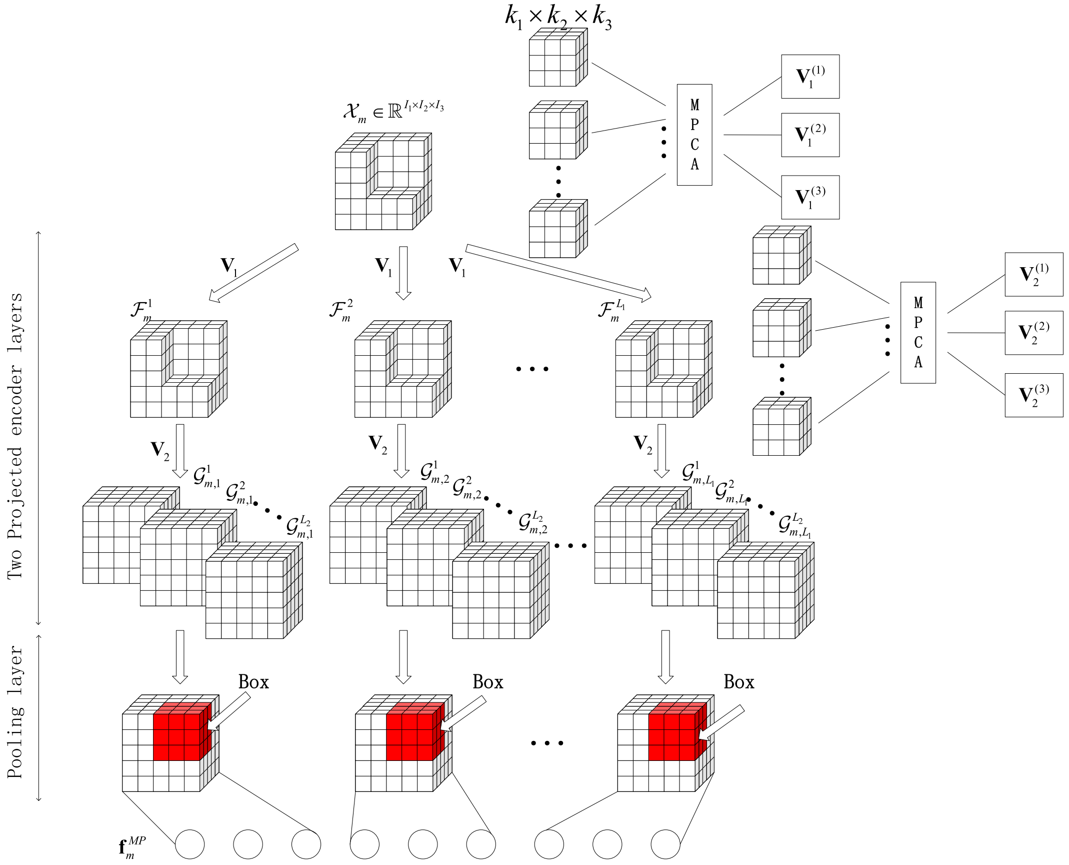


FIGURE 3. The architecture of two-layer MPCANet (MPCANet-2).

B. THE SECOND STAGE OF MPCANet-2

Similar to the first stage, we collect all the overlapping tensor patches of $\{\mathcal{F}_m^{l_1} \in \mathbb{R}^{I_1\times I_2\times I_3}, l_1 = 1,\ldots,L_1\}$, and form a new set of tensor patches $\mathcal{A}_m^2 = \{\mathcal{A}_{m,l,q}^2 \in \mathbb{R}^{k_1\times k_2\times k_3}, l_1 = 1,\ldots,L_1; q = 1,\ldots,I_1I_2I_3\}$, where $\mathcal{A}_{m,l,q}^2$ denotes the qth tensor patch of the tensor $\mathcal{F}_m^{l_1}$. By repeating this procedure for all the tensor features, we get $\mathcal{A}^2 = \{\mathcal{A}_m^2, m = 1,\ldots,M\}$. By exploiting MPCA algorithm [19], we can get a set of projection matrices $V_2 = \{\mathbf{V}_2^{(n)} \in \mathbb{R}^{k_n\times p_n}, n = 1,2,3\}$, which are then used to project $\mathcal{A}_m^2$ to $\mathcal{S}_m^2 = \{\mathcal{S}_{m,l_1,q}^2 \in \mathbb{R}^{p_1\times p_2\times p_3}, l_1 = 1,\ldots,L_1; q = 1,\ldots,I_1I_2I_3\}$. Each element of $\mathcal{S}_m^2$ is converted to a vector, and then a new set of vectors $\mathbf{z}_m^2 = \{\mathbf{z}_{m,l_1,q}^2 \in \mathbb{R}^{p_1p_2p_3}, l_1 = 1,\ldots,L_1; q = 1,\ldots,I_1I_2I_3\}$ is obtained. Picking up the first L_2 elements of $\mathbf{z}_{m,l_1,q}^2$, a new vector $\hat{\mathbf{z}}_{m,l_1,q}^2 \in \mathbb{R}^{L_2}$ is formed. Subsequently, by constructing the same matrix for all projected tensor patches of $\mathcal{F}_m^{l_1}$ and putting them together, a matrix may be generated as

$$\hat{\mathbf{z}}_{m,l_1}^2 = \begin{bmatrix} \hat{\mathbf{z}}_{m,l_1,1}^2 & \hat{\mathbf{z}}_{m,l_1,2}^2 & \cdots & \hat{\mathbf{z}}_{m,l_1,L_2}^2 \end{bmatrix}^T \in \mathbb{R}^{I_1I_2I_3\times L_2}. \tag{17}$$

The columns of $\hat{\mathbf{z}}_{m,l}^2$ is then mapped back to a set of tensor objects $\left\{\mathcal{G}_{m,l_1}^{l_2} \in \mathbb{R}^{I_1\times I_2\times I_3}, l_2 = 1,\ldots,L_2\right\}$, where $\mathcal{G}_{m,l_1}^{l_2}$ is the l_2th tensor feature map of $\mathcal{F}_m^{l_1}$ in the second stage. Note that the number of outputs of the second stage is L_1L_2. One can simply repeat the above process to construct more stages if a deeper architecture is found to be necessary.

C. THE OUTPUT STAGE OF MPCANet-2

Each tensor feature $\left\{\mathcal{G}_{m,l_1}^{l_2} \in \mathbb{R}^{I_1\times I_2\times I_3}, l_2 = 1,\ldots,L_2\right\}$ is binarized by using Heaviside step function $H(\cdot)$, the value of which is one for positive entries and zero otherwise. $\left\{\tilde{\mathcal{G}}_{m,l_1}^{l_2} \in \mathbb{R}^{I_1\times I_2\times I_3}, l_2 = 1,\ldots,L_2\right\}$ denotes these b inarized tensor features. Since different tensor features can describe different variations of the original tensor data, $\left\{\tilde{\mathcal{G}}_{m,l_1}^{l_2}, l_2 = 1,\ldots,L_2\right\}$ should be weighted to form new single tensor features as follows [7]:

$$\mathcal{W}_m^{l_1} = \sum_{l_2=1}^{L_2} 2^{l_2-1}\tilde{\mathcal{G}}_{m,l_1}^{l_2}. \tag{18}$$

Note that each element of the tensor feature $\mathcal{W}_m^{l_1}$ is an integer in the range of $[0, 2^{L_2-1}]$.

Next, a spatial pooling operation is applied to $\mathcal{W}_m^{l_1}$. The cubic tensor feature is divided into B boxes. We compute the histogram of the decimal values for each box and denote it as $\text{hist}(Box)_b, b = 1,\ldots,B$. After the above pooling process,

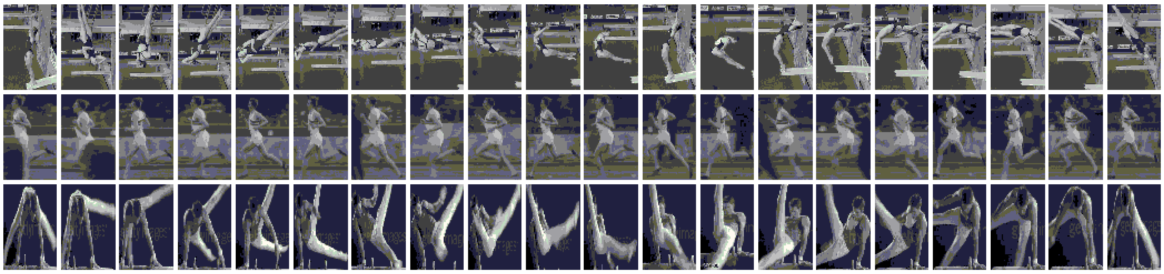


FIGURE 4. Some frames from the UCF sports action dataset. The top images are a series of diving sports. The middle images are a series of running sports. The bottom images are a series of gymnastic sports.

we simply concatenate all the histograms of B boxes into one vector, obtaining

$$\mathbf{f}_m^{MP} = \left[\mathrm{hist}(Box)_1 \quad \mathrm{hist}(Box)_2 \quad \cdots \quad \mathrm{hist}(Box)_B\right] \in \mathbb{R}^{(2^{L_2})L_1 B}. \tag{19}$$

Boxes can either be overlapping or non-overlapping in MPCANet, depending on the application [7]. The feature vector is then sent to a SVM classifier [25]–[27]. Note that the feature vector is in general of very high dimension but very sparse therefore, the performance of SVM classifier is much better than that of k-nearest neighbor (KNN) classifier [16].

D. THE COMPUTATIONAL COMPLEXITY OF MPCANet-2

In this subsection, we examine the computational complexity of MPCANet-2 for each third-order tensor object in a similar manner to that of PCANet-2 introduced in [7]. In the first stage of MPCANet-2, forming the patch-mean-removed tensor version of $\mathcal{A}_{m,q}^1 \in \mathbb{R}^{k_1\times k_2\times k_3}$ costs $k_1k_2k_3 + k_1k_2k_3I_1I_2I_3$ flops. Note that this step is implied in the preprocessing of MPCA algorithm [19]. The following usage of MPCA algorithm consists of the formation of the matrices $\mathbf{\Phi}^{(n)}$ in (5) and the eigen-decomposition process, which have the complexities of $O(k_1^2k_2k_3I_1I_2I_3)$ and $O((k_1k_2k_3)^3)$. The MPCA filter convolution requires $L_ik_1k_2k_3I_1I_2I_3$ flops for stage i. In the output layer, the conversion of L_2 binary bits to a decimal number costs $2L_2I_1I_2I_3$, and the naive histogram operation is of complexity $O(I_1I_2I_3BL_2\log 2)$. Therefore, the overall complexity of MPCANet-2 is $O(k_1k_2k_3I_1I_2I_3(L_1 + L_2) + (k_1k_2k_3)^3 + k_1^2k_2k_3I_1I_2I_3)$ when assuming $I_1I_2I_3 \geq \max(k_1, k_2, k_3, L_1, L_2, B)$.

V. EXPERIMENTAL RESULTS

This section gives three applications of the proposed MPCANet for tensor object classification. Although there are many variants of PCANet, most of them are optimized for a particular situation. For example, parameters of weighted-PCANet [13] and Deep learning PCANet (DLPCANet) [15] are optimized for face recognition and human age estimation. The optimized parameters of [13] and [15] are not suitable for tensor object classification. Therefore, we only compare our proposed MPCANet with PCANet [7] and also MPCA algorithm [19].

A. PERFORMANCE ON THE UCF SPORTS ACTION DATABASE

The UCF database [28] is a sports action video database; it contains 9 classes and approximately 200 videos of sports action at a resolution of 720 × 480. The videos are typically chosen from broadcast television channels such as BBC and ESPN. Some frames taken from the videos in the UCF sports action dataset are shown in Fig. 4.

The video samples of the UCF database need to be preprocessed before performing the proposed MPCANet. We first convert each video frame from color space to gray scale one and then manually extract the sportsman in each frame with a size of 400 × 250. The cropped video frames are then resized to 80 × 50 pixels to reduce computational complexity. The dimension of the third-mode (or time mode) is set to 20 to maintain a complete sports action. When preprocessing is complete, video samples can be represented by a set of tensor objects $\{\mathcal{X}_m \in \mathbb{R}^{80\times 50\times 20}, m = 1, \ldots, M\}$. We randomly pick up half of the tensor objects in each class as a training set and the others as a testing set.

Since MPCA [19] is the tensor extension of the conventional PCA, we use parameters in the proposed MPCANet consistent with those of PCANet reported in [7]. For computational simplicity, we set the size of the third-mode dimension of the tensor patch to 20, which is equal to that of each tensor object. That is to say, the tensor patch sizes in MPCANet are set to $3\times3\times20$, $5\times5\times20$ and $7\times7\times20$. Thus, the output tensor feature in the second stage of MPCANet-2 is the second-order tensor (or matrix). We set the number of filters to $L_1 = L_2 = 8$ in the convolution layer and the box sizes to multiples of $8 \times 5 \times 1$ in the pooling layer of MPCANet. To offer some degree of translation invariance in tensor objects, the overlapping ratio of the box is set to 50%. For PCANet, we use the parameters recommended by the authors [7]. The patch sizes are 3×3, 5×5 and 7×7. The number of filters is 8. The block sizes are multiples of 8×5 and the overlapping ratio of the block is 50%.

In addition to PCANet, we also compare the proposed MPCANet with the conventional MPCA [19], where liblinear SVM is used to do the classification. The recognition rates of MPCANet, PCANet [7] with different patch sizes and block sizes are shown in Fig. 5. The best recognition accuracy rates

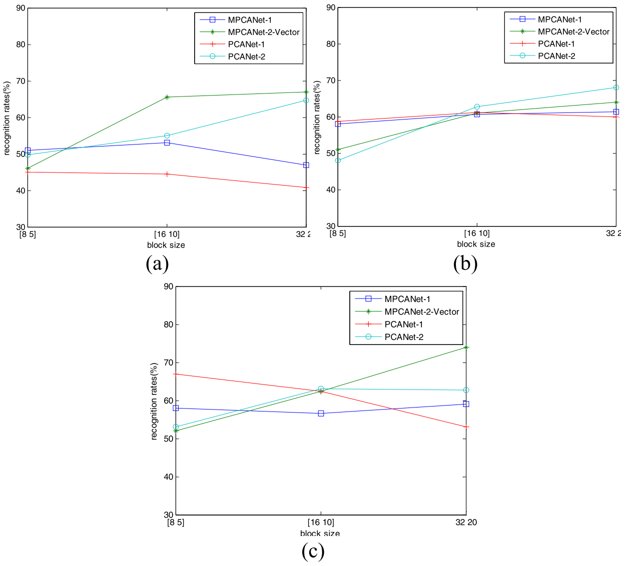


FIGURE 5. The recognition rates of MPCANet and PCANet for different patch sizes $k_1 \times k_2$. (a) 3 × 3. (b) 5 × 5. (c) 7 × 7.

of MPCANet, PCANet and MPCA are listed in Table 1.

TABLE 1. The best performance of MPCANet, PCANet [7], and MPCA [19] on the UCF sports action database.

Methods	Accuracy (%)
MPCANet-1	63.01
MPCANet-2	**73.93**
PCANet-1 [7]	67.12
PCANet-2 [7]	68.49
MPCA [19]	42.34

It can be observed that both one-layer MPCANet (MPCANet-1) and one-layer PCANet (PCANet-1) outperform MPCA. May be the reason is that MPCA algorithm does not have a nonlinear pooling layer, leading to less robust features for visual content compared to MPCANet-1 and PCANet-1. PCANet-1 achieves the best performance among one-layer networks but the improvement from PCANet-1 to PCANet-2 is not as large as that of MPCANet. One can see that MPCANet-2 obtains the best recognition result in two-layer networks since it utilizes the information of spatial structures of tensor objects much more efficiently than PCANet-2.

B. TESTING ON THE UCF11 DATABASE

In this subsection, we test the performance of MPCANet on the UCF11 database [29] for tensor object classification. The UCF11 database contains 11 action categories: basketball shooting, biking, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and dog walking. Some examples in the UCF11 dataset are shown in Fig. 6. The video samples are all 240 × 320 pixels and have been manually collected from YouTube. For each category, the video samples belong to 25 groups with more than 4 action clips in each group. We only choose the first ten groups in each category to test the performance of MPCANet, PCANet [7], and MPCA [19]. The total number of experimental video samples is 642. For each group, half of the video samples are randomly selected for training, and the others are used for testing. We normalize the video samples first to reduce the differences in frame variation. For videos whose frames are larger than 20, we only choose the first twenty frames. For videos whose frames are less than 20, we simply copy the last frame to fill them. Every video sample is resized to 48 × 64 to reduce the calculation burden. Therefore, the video samples of the UCF11 database are represented by a set of tensor objects $\{\mathcal{X}_m \in \mathbb{R}^{48\times 64\times 20}, m = 1, \ldots, M\}$.

TABLE 2. The best performance of MPCANet, PCANet [7], and MPCA [19] on the UCF11 database.

Methods	Accuracy (%)
MPCANet-1	59.53
MPCANet-2	**79.26**
PCANet-1 [7]	58.68
PCANet-2 [7]	76.92
MPCA [19]	45.15

The patch sizes and the overlapping ratios of MPCANet and PCANet are the same as the previous experiment on the UCF sports action database. Three box sizes are considered here: 6 × 8, 12 × 16 and 24 × 32. Table 2 reports the best performance of MPCANet, PCANet, and MPCA. We observe that MPCANet-2 still outperforms the other methods, and MPCA is still worst in terms of classification accuracy.

C. PERFORMANCE IN A MEDICAL IMAGE DATABASE

Medical image processing has attracted a great deal of attention [30]–[36]. In this subsection, the experiments are carried out on the classification of the medical image database that we collected. The database contains different samples of abdomens, lumber, brains, lungs, hearts and chests, and each class has 48 samples, all of which are three-dimensional (3-D) image sequences. The samples are taken from CT or MRI images. Some examples of the dataset are shown in Fig. 7. The lung sequence of images consist of 50 low-dose documented whole-lung CT scans. The CT scans were obtained in a single breath hold with a 1.25 mm slice thickness, and the original size of the flames is 512 × 512. Both the brain images and the abdomen images are MRI images; they are 260 × 260 slices with 0.6 mm thickness.

It is necessary to preprocess the images in the database before performing the proposed MPCANet. First, we resize all the frames to 64 × 64 pixels to reduce computational complexity. The dimension of the third-mode is set to 20. After preprocessing, all the samples can be represented as a set of tensor objects $\{\mathcal{X}_m \in \mathbb{R}^{64\times 64\times 20}, m = 1, \ldots, M\}$.

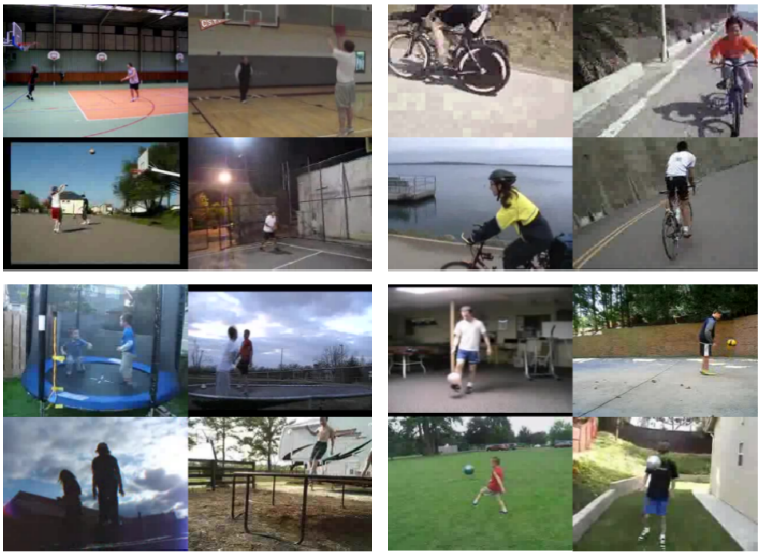


FIGURE 6. Some examples from the UCF11 dataset. The top left corner block shows 4 basketball scenes. The top right corner shows biking. Trampoline activities are shown at bottom left. In the right bottom, soccer is shown.

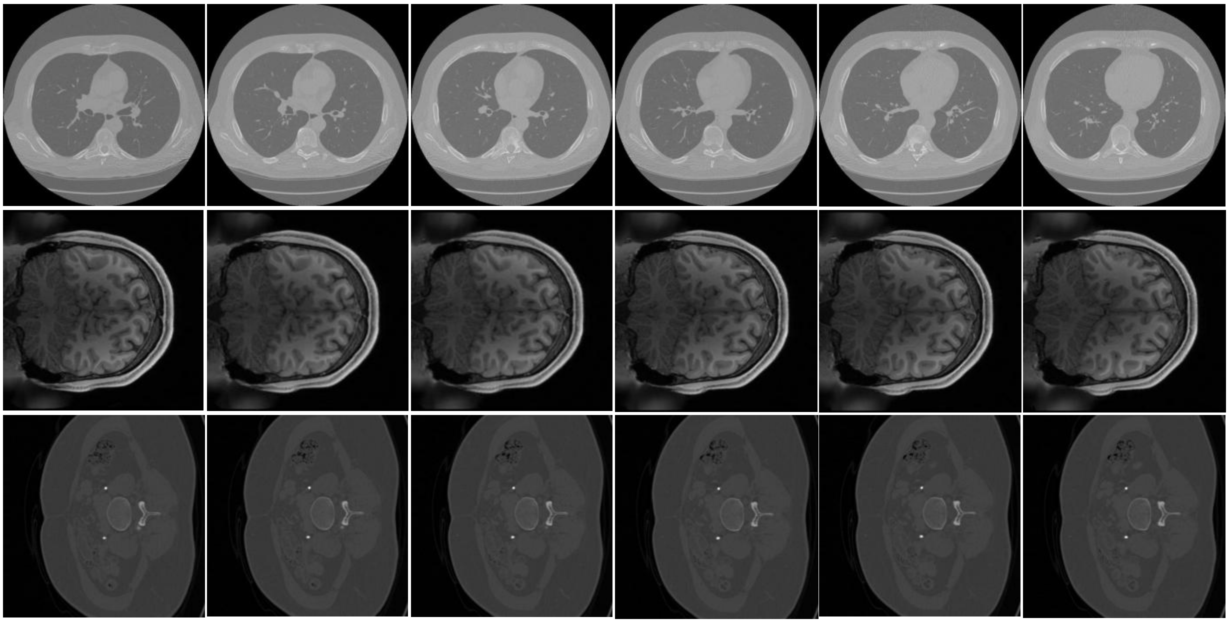


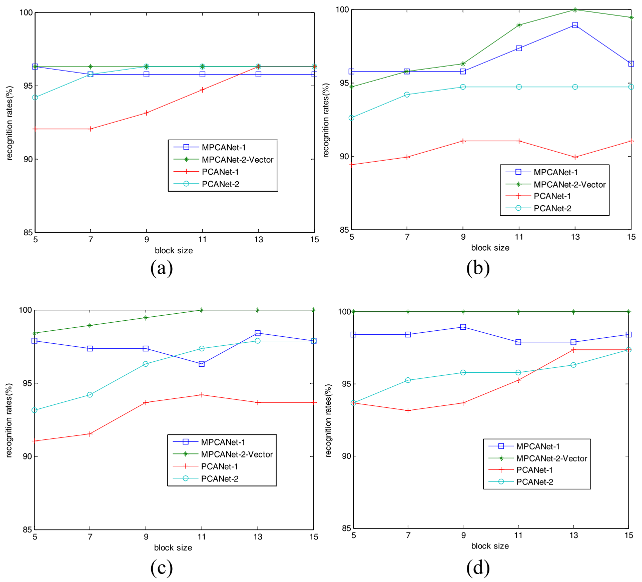
FIGURE 7. Some examples from the medical image database. The top images are a series of lung images. The middle images are a series of brain images. The bottom images are a series of abdomen images.

For each class, we randomly select half of the tensor objects for training and test on the others.

In addition to PCANet [7], we also compare the proposed MPCANet with conventional MPCA [19]. The recognition rates of MPCANet, PCANet [7] with different patch sizes and block sizes are shown in Fig. 8.

It is apparent that with the increase in patch size, the results of these four methods tend to be better. However, the performance does not improve greatly when the patch size is larger than 7 × 7. In addition, the recognition rates of the learning networks gradually improve as the block size increases. From the figure, we can see that the performance of the MPCANet is somewhat less sensitive to block size. To take MPCANet-2 as an example, when the patch size is set to 9 × 9, the recognition rate is almost 100%, regardless of the changes in block size.

The best recognition accuracy rates of MPCANet, PCANet and MPCA are listed in Table 3. We note that MPCANet-2 still outperforms the other methods and MPCA is still the worst in terms of classification accuracy. It is also shown that



(a) (b)

(c) (d)

FIGURE 8. The recognition rates of MPCANet and PCANet for different patch sizes $k_1 \times k_2$. (a) 3 × 3. (b) 5 × 5. (c) 7 × 7. (d) 9 × 9.

TABLE 3. The best accuracy rates of MPCANet, PCANet [7], and MPCA[19] on the medical image database.

Methods	Accuracy (%)
MPCANet-1	98.41
MPCANet-2	**100**
PCANet-1 [7]	96.30
PCANet-2 [7]	97.88
MPCA [19]	49.21

one-layer MPCANet (MPCANet-1) outperforms a two-layer PCANet (PCANet-2) in this experiment. Maybe the reason is that one-layer MPCA captures better intra-class variability than two-layer PCA on this medical image database.

VI. CONCLUSIONS

In this paper, we propose and implement a novel deep learning architecture, that is, MPCANet. Compared to PCANet, MPCANet involves tensor interactions among stages to protect the spatial structures of multi-dimensional objects, which efficiently improve its recognition rate. MPCANet is composed of projection dictionaries, the projection encoder layer, and the pooling layer. An approach to mapping the tensor objects to a corresponding feature vector and the architecture of MPCANet has been described in detail.

We have executed the MPCANet on the UCF sports action database, the UCF11 database, and a medical image dataset to verify the effectiveness of MPCANet. The experimental results demonstrate that MPCANet outperforms the conventional PCANet and MPCA in tensor object classification. It provides deep insight on tensor extension on ConvNets.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Dec. 2012, vol. 25. no. 2, pp. 1097–1105.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. (2013). "OverFeat: Integrated recognition, localization and detection using convolutional networks." [Online]. Available: https://arxiv.org/abs/1312.6229

[4] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[5] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.

[6] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

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

[8] Z. Feng, L. Jin, D. Tao, and S. Huang, "DLANet: A manifold-learning-based discriminative feature learning network for scene classification," *Neurocomputing*, vol. 157, pp. 11–21, Jun. 2015.

[9] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," *Neurocomputing*, vol. 187, pp. 49–58, Apr. 2016.

[10] Y. Zhao, R. Wang, W. Wang, and W. Gao, "Multilevel modified finite radon transform network for image upsampling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2189–2199, Dec. 2015.

[11] Z. Lei, D. Yi, and S. Z. Li, "Learning stacked image descriptor for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1685–1696, Sep. 2015.

[12] J. Shi, J. Wu, Y. Li, Q. Zhang, and S. Ying, "Histopathological image classification with color pattern random binary hashing based PCANet and matrix-form classifier," *IEEE J. Biomed. Health Inform.*, doi: 10.1109/JBHI.2016.2602823.

[13] J. Huang and C. Yuan, "Weighted-PCANet for face recognition," in *Proc. Int. Conf. Neural Inf. Process.*, Istanbul, Turkey, Nov. 2015, pp. 246–254.

[14] W. L. Hao and Z. Zhao, "Incremental PCANet: A lifelong learning framework to achieve the plasticity of both feature and classifier constructions," in *Proc. 8th Int. Conf. Brain Inspired Cognit. Syst.*, Beijing, China, Nov. 2016, pp. 298–309.

[15] D. Zheng, J. Du, W. Fan, J. Wang, and C. Zhai, "Deep learning with PCANet for human age estimation," in *Proc. Int. Conf. Intell. Comput.*, Lanzhou, China, Aug. 2016, pp. 300–310.

[16] R. Zeng *et al.*, "Color image classification via quaternion principal component analysis network," *Neurocomputing*, vol. 216, pp. 416–428, Dec. 2016.

[17] J. Wu, J. Shi, S. Ying, Q. Zhang, and Y. Li, "Learning representation for histopathological image with quaternion Grassmann average network," in *Proc. Int. Workshop Mach. Learn. Med. Imag.*, Athens, Greece, Oct. 2016, pp. 122–129.

[18] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*. London, U.K.: Springer, 2011, pp. 19–49.

[19] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[20] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, "Scalable and compact representation for motion capture data using tensor decomposition," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 255–259, Mar. 2014.

[21] Z.-Z. Yu, C.-C. Jia, W. Pang, C.-Y. Zhang, and L.-H. Zhong, "Tensor discriminant analysis with multiscale features for action modeling and categorization," *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 95–98, Feb. 2012.

[22] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1944–1957, Aug. 2013.

[23] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 388–396, Feb. 2013.

[24] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Aug. 2008.

[26] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1403–1416, Jul. 2015.

[27] B. Gu and V. S. Sheng, "A robust regularization path algorithm for ν-support vector classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2527796.

[28] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[29] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos 'in the wild,'" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1996–2003.

[30] Y. Chen *et al.*, "Thoracic low-dose CT image processing using an artifact suppressed large-scale nonlocal means," *Phys. Med. Biol.*, vol. 57, no. 9, pp. 2667–2688, 2012.

[31] Y. Chen *et al.*, "Improving low-dose abdominal CT images by weighted intensity averaging over large-scale neighborhoods," *Eur. J. Radiol.*, vol. 80, no. 2, pp. e42–e49, 2011.

[32] Y. Chen *et al.*, "Bayesian statistical reconstruction for low-dose X-ray computed tomography using an adaptive-weighting nonlocal prior," *Comput. Med. Imag. Graph.*, vol. 33, no. 7, pp. 495–500, 2009.

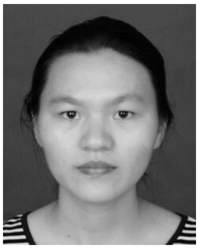
[33] Y. Zhang, Z. Dong, S. Wang, G. Ji, and P. Phillips, "Prediction of MCI to Alzheimer's conversion based on tensor-based morphometry and kernel support vector machine," *Alzheimer's Dementia*, vol. 11, no. 7, p. 702, 2015.

[34] Y. Zhang, Z. Dong, S. Wang, G. Ji, and P. Phillips, "Prediction of Alzheimer's disease and mild cognitive impairment based on structural volumetric MR images by 3D discrete wavelet transform and artificial neural network," *Alzheimer's Dementia*, vol. 11, no. 7, pp. P78–P79, 2015.

[35] X. Qu, Y. Hou, F. Lam, D. Guo, J. Zhong, and Z. Chen, "Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator," *Med. Image Anal.*, vol. 18, no. 6, pp. 843–856, 2014.

[36] Y. Liu, Z. Zhan, J.-F. Cai, D. Guo, Z. Chen, and X. Qu, "Projected iterative soft-thresholding algorithm for tight frames in compressed sensing magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 35, no. 9, pp. 2130–2140, Sep. 2016.

JIASONG WU (M'09) received the B.S. degree in biomedical engineering from the University of South China, Hengyang, China, in 2005, and the joint Ph.D. degree from the Laboratory of Image Science and Technology (LIST), Southeast University, Nanjing, China, and the Laboratoire Traitement du signal et de l'Image, University of Rennes 1, Rennes, France, in 2012. He is currently with LIST, as a Lecturer. His research interest mainly includes deep learning, fast algorithms of digital signal processing, and its applications. He received the Eiffel Doctorate Scholarship of Excellence in 2009 from the French Ministry of Foreign Affairs and also the Chinese Government Award for outstanding Self-Financed Student Abroad in 2010 from the China Scholarship Council.

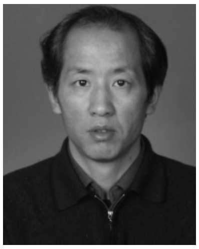


SHIJIE QIU received the B.S. degree in mathematical science from Soochow University in 2015. She is currently pursuing the M.S. degree in computer science, Southeast University. Her current research interests include in deep learning and pattern recognition.

RUI ZENG received the B.S. degree in computer science from the Wuhan Institute of Technology in 2012 and the M.S. degree in computer science from Southeast University in 2015. He is currently pursuing the Ph.D. degree in computer science with the Queensland University of Technology. His research interests include deep learning and pattern recognition.

YOUYONG KONG received the B.S. and M.S. degrees in computer science and engineering from Southeast University, Nanjing, China, in 2008 and 2011, respectively, and the Ph.D. degree in imaging and diagnostic radiology from The Chinese University of Hong Kong, Hong Kong, in 2014. He is currently an Assistant Professor with the College of Computer Science and Engineering, Southeast University. His current research interests include machine learning, and medical image processing and brain network analysis.

LOTFI SENHADJI (M'95–SM'99) received the Ph.D. degree in signal processing and telecommunications from the University of Rennes 1, Rennes, France, in 1993. He is currently a Professor and the Head of the INSERM Research Laboratory LTSI. His is also the Co-Director of the French-Chinese Laboratory, Centre de Recherche en Information Biomédicale Sino-Français. He has authored over 100 research papers in journals and conferences, and he contributed to five handbooks. His main research efforts are focused on signal processing with particular emphasis on event detection, classification, and interpretation of signals. He is a Senior Member of the IEEE EMBS and the IEEE Signal Processing Society.



HUAZHONG SHU (M'00–SM'06) received the B.S. degree in applied mathematics from Wuhan University, China, in 1987, and the Ph.D. degree in numerical analysis from the University of Rennes 1, Rennes, France, in 1992. He is currently a Professor of the LIST Laboratory and the Co-Director of the CRIBs. His recent work concentrates on the image analysis, pattern recognition and fast algorithms of digital signal processing. He is a Senior Member of IEEE Society.