International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.8 (2016), pp.275-284
http://dx.doi.org/10.14257/ijsip.2016.9.8.24

Tensorial Feed Forward Neural Networks with Random Weights for Gait Recognition Using MPCA Features

Fan Zhang[1*] and Ke Sun[2]

[1]*School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou, 450045, China*
[2]*Mathematics and Computational Science College, Guilin University of Electronic Technology, Guilin, 541000, China*
zfgh8221@163.com

Abstract

Because of gait sequences are naturally three-dimensional data, there have been several tensorial feature extraction methods to deal with tensors while there are effective tensorial classifiers. In this work, by using a linear tensor projection, a new classifier based on neural networks with random weights is introduced. Due to the proposed algorithm can classify gait samples directly without vectorizing them, the intrinsic structure information of the input data can be reserved. In addition, discriminative features sets are generated using MPCA to ascertain classification accuracy. Finally, Extensive experiments are carried out on two gait databases and results are compared against state-of-the-art techniques. It is demonstrated that the proposed algorithm MPCA plus TNNRE achieves better recognition performance.

Keywords*: Gait recognition, Multilinear principal component analysis (MPCA), Feature extraction, Neural networks with random weights (NNRW), Classification*

1. Introduction

During the past decades, there is an increasing demand for automated person identification at a distance in controlled environments such as banks, parking lots, airports, car parks and large civic structures. Different from the traditional biometrics that to be required in close distance, such as iris, face, fingerprint, and palm print, gait information can be available at high enough resolution for recognition. As the style of walking of a person, gait recognition focus on the identification of individuals in video sequences by the way they walk [1-2]. In gait recognition systems, binary gait silhouettes are always taken as the input source data. The sequences can be regard as three-order tensors with the spatial row, column and the temporal modes for the three dimensions [3]. In order to deal with these tensor objects directly, the traditional vector based linear subspace learning method such as Principal Component Analysis (PCA) [4] and Linear Discriminant Analysis (LDA) [5] should be modified. Since PCA and LDA should firstly reshape these tensorial data into vectors in a very high dimensional space. It is generally accepted that this reshaping breaks the potential spatial structure of the original data and leads high computational burden. To address these issues, the development of multilinear subspace learning approaches operating directly on the gait sequences has been motived. For example, Multilinear principal component analysis (MPCA) [6], a tensor version of PCA, applies PCA transformation on each mode of tensors while multilinear discriminant analysis (MDA) [7] applies LDA to transform each dimensionality of tensors. Hence, the natural structure of the gait sequences will be reserved and a better performance will be achieved.

* Corresponding Author

ISSN: 2005-4254 IJSIP

Nevertheless, for the classification step, the classical image classifiers with good generalization include nearest neighbor (NN), artificial neural network (ANN) [8], support vector machine (SVM) [9], and so on. With the same limitation, traditional classification methods are usually vector-based, such as SVM and ANN. When they are employed to classify an input image, we have to transform matrix data into vector data at first. Besides, due to the distance between two data with form either tensor or vector is the same, NN classifier can be used to classify the feature matrices directly. Therefore, NN classifier is usually applied after many multilinear dimension reduction methods just as in [10]. However, the structure of NN classifier is too simple to obtain satisfying recognition rate. Consequently, it is desirable to propose a novel classifier which can classify matrix data directly and to preserve the intrinsic structure effectively. In order to construct the classifier, a kind of special feed forward networks be employed that introduced first in [11], named neural networks with random weights (NNRW). Because NNRW randomly assigns the input weights and the bias of neurons in the hidden layer, these networks have fast learning speed and perfect classification performance [12]. As a tensor extension of NNRW, we introduce a novel tensor based classification method named as tensorial neural networks with random weights (TNNRW) for tensor objects classification. Without converting tensor objects into vectors, TNNRW can classify them directly. TNNRW not only takes the advantage of NNRW, but also preserves the natural structure of the input data. Furthermore, with fewer parameters to be calculated, a faster computing speed can be achieved.

For most practical pattern recognition systems, feature extraction should retain most of the useful information in input data while keeping the dimension of the features as low as possible. In this paper, MPCA is used for feature extraction from original gait sequences with a series of projection matrices to maximize the captured variation. Then the extracted features are put into the proposed TNNRW for classify. Several experiments on different gait databases are presented for illustration.

The remainder of the paper is organized as follows. Section 2, introduces some basic tensor algebra. Section 3, provides a brief introduction of MPCA, NNRW, and then the new classifier TNNRW is summarized in detail. In Section 4, we analyze experiment results on gait databases to verify the properties of the proposed method and compares performance against the other algorithms. Finally, the major findings and conclusions are drawn in Section 5.

2. Tensor Fundamentals

The elements of a tensor are to be addressed by a number of indices that are used to define the order of the tensor object. Notably, each index defines a "mode" [13]. Following the notation in [14], we denote vectors by lowercase boldface letters, *e.g.*, **x**; matrices by uppercase boldface, *e.g.*, **U**; and tensors by calligraphic letters, *e.g.*, $\mathcal{A}$. Tensor is a generalization of vector and matrix as vectors are first-order tensors, and matrices are second-order tensors. An *N*th-order tensor is denoted as $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Their elements are addressed by *N* indices i_n , $n = 1, \cdots, N$, and each i_n addresses the mode-*n* of $\mathcal{A}$.

The mode-*n* unfolding of $\mathcal{A}$ is defined as the I_n dimensional vectors are denoted as

$$\mathcal{A}_{(n)} \in \mathbf{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)} \tag{1}$$

Where the column vectors of $\mathcal{A}_{(n)}$ are gained from $\mathcal{A}$ by varying its index i_n while keeping all the other indices fixed.

The mode-*n* product of a tensor A by a matrix $\mathrm{U} \in \mathbf{R}^{J_n \times I_n}$, denoted by $\mathrm{A} \times_n \mathrm{U}$, is a tensor defined as

$$(\mathrm{A} \times_n \mathrm{U})(i_1, \cdots, i_{n-1}, j_n, i_{n+1}, \cdots, i_N) = \sum_{i_n} \mathrm{A}(i_1, i_2, \cdots, i_N) \cdot \mathrm{U}(j_n, i_n) \tag{2}$$

One of the most commonly used tensor decompositions is Tucker, which can be regarded as higher-order generalization of the matrix Singular Value Decomposition (SVD). Let $\mathrm{A} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ denotes a *N*th-order tensor, then the Tucker decomposition is defined as follows

$$\mathrm{A} = \mathrm{S} \times_1 \mathrm{U}^{(1)} \times_2 \mathrm{U}^{(2)} \cdots \times_N \mathrm{U}^{(N)} \tag{3}$$

Where $\mathrm{S} \in \mathbf{R}^{P_1 \times P_2 \times \cdots \times P_N}$ with $P_n < I_n$, denotes the core tensor and $\mathrm{U}^{(n)} = \left[\mathrm{u}_1^{(n)} \mathrm{u}_2^{(n)} \cdots \mathrm{u}_{P_n}^{(n)} \right]$ is an $I_n \times P_n$ matrix. The scalar product of two tensors $\mathrm{A}, \mathrm{B} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as

$$\langle \mathrm{A}, \mathrm{B} \rangle = \sum_{i_1} \cdots \sum_{i_N} \mathrm{A}(i_1, i_2, \cdots, i_N) \cdot \mathrm{B}(i_1, i_2, \cdots, i_N) \tag{4}$$

The Frobenius norm of $\mathcal{A}$ is defined as

$$\|\mathrm{A}\| = \sqrt{\langle \mathrm{A}, \mathrm{A} \rangle} = \|\mathrm{A}_{(n)}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} a_{i_1 i_2 \cdots i_N}^2} \tag{5}$$

3. Multilinear PCA and Tensorial NNRW

3.1. Multilinear Principal Component Analysis

As a multilinear extension of PCA, MPCA algorithm which is firstly introduced in [6] seeks a multilinear projection that transform the input data from a tensor space to another (low-dimensional) tensor space. A set of M tensor object samples $\{\mathrm{X}_1, \mathrm{X}_2, \cdots, \mathrm{X}_M\}$ are available for training and each tensor sample $\mathrm{X}_m \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The aim of MPCA is to determine the *N* projection matrices $\{\mathrm{U}^{(n)} \in \mathbf{R}^{I_n \times P_n}, n = 1, \cdots, N\}$ to maximize the total tensor scatter Ψ_{Y} of the extracted low-dimensional features:

$$\{\tilde{\mathrm{U}}^{(n)}\} = \arg\max_{\{\mathrm{U}^{(n)}\}} \Psi_{\mathrm{Y}} = \arg\max_{\{\mathrm{U}^{(n)}\}} \sum_{m=1}^{M} \|\mathrm{Y}_m - \bar{\mathrm{Y}}\|_F^2 \tag{6}$$

Where $\mathrm{Y}_m = \mathrm{X}_m \times \{\mathbf{U}^{(n)^T}\}_{n=1}^{N}$, $\bar{\mathrm{Y}} = \frac{1}{M} \sum_{m=1}^{M} \mathrm{Y}_m$.

The *N* optimization subproblems are solved by finding the mode-*n* projection matrix $\mathbf{U}^{(n)}$ that maximizes the mode-*n* total scatter conditioned on the projection matrices in all the other modes. The P_n eigenvectors reside in the matrix $\mathbf{U}^{(n)}$ corresponding to the largest P_n eigenvalues of the matrix $\mathbf{\Phi}^{(n)}$:

$$\mathbf{\Phi}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{C}_m^{(n)} \mathbf{C}_m^{(n)^T} \right) \tag{7}$$

$$\mathbf{C}_m^{(n)} = \left(\mathbf{X}_{m(n)} - \overline{\mathbf{X}}_{(n)} \right) \times_1 \mathbf{U}^{(1)^T} \times \cdots \times_{n-1} \mathbf{U}^{(n-1)^T} \times_{n+1} \mathbf{U}^{(n+1)^T} \times \cdots \times_N \mathbf{U}^{N^T} \tag{8}$$

Where $\mathbf{X}_{m(n)}$ is the mode-*n* unfolding of $\mathbf{X}_m$, $\overline{\mathbf{X}}_{(n)}$ is the mode-*n* mean matrix.

3.2. A Brief Review of NNRW

Feed-forward neural networks are ideal classifiers for approximating complex nonlinear mappings directly from the input data. Among them, Single-hidden Layer Feed-forward Neural Network （SLFNN）[15] has very strong learning ability and has been applied in many fields. However, due to the hidden layer parameters and the output weights need to be trained and tuned properly based on the input samples, the learning speed of SLFNN is too slow to meet the demand in actual situations. As a kind of special learning method for SLFNN, NNRW can randomly set the input weights, hidden layer biases, and the whole process of NNRW does not need iteration and can obviously improve the neural network learning speed.

Suppose a set of *N* arbitrary distinct samples $\left(\mathbf{x}_j, \mathrm{t}_j\right)$, where $\mathbf{x}_j = \left[x_{j1}, x_{j2}, \cdots, x_{jn} \right]^T \in \mathbf{R}^n$ and $\mathbf{t}_j = \left[t_{j1}, t_{j2}, \cdots, t_{jm} \right]^T \in \mathbf{R}^m$. $\mathbf{t}_j$ is the class label of the input data, if $\mathbf{x}_j$ belong to the class *m*, then $t_{jm} = 1$ while other parameters in $\mathbf{t}_j$ is 0. A typical FNN with single hidden layer can be commonly modeled as:

$$\mathrm{y}_j = \sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathrm{x}_j\right) = \sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathbf{w}_i^T \cdot \mathrm{x}_j + \mathrm{b}_i\right), \quad j = 1, \cdots, N \tag{9}$$

Where L is the number of hidden nodes, $g(\cdot)$ is the active function, $\mathbf{w}_i = [w_{i1}, w_{i2}, \cdots, w_{in}]$ is the weight vector connecting the *i*-th hidden node and the input nodes, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the *i*-th hidden node and the output nodes, and b_i is the bias of the *i*-th hidden node. Figure 1, shows the architecture of a complete NNRW process.

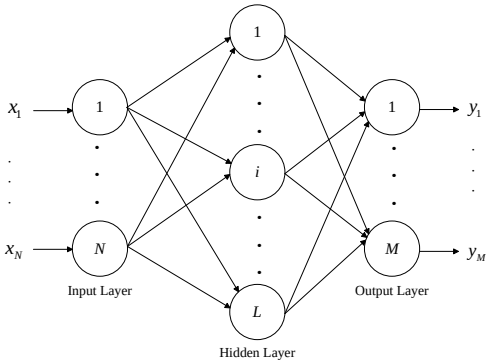

Figure 1. Architecture of a Complete NNRW Classifier

Considering $\mathbf{t}_j$ is the corresponding observation value. The NNRW reliably approximates *N* samples with minimum error:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathbf{w}_i^T \cdot \mathrm{x}_j + \mathrm{b}_i\right) = \mathbf{t}_j, \quad j = 1, \cdots, N \tag{10}$$

According to NNRW proposed in [16], each element of the input weights and biases is selected randomly, then the output weights can be calculated by using Moore-Penrose generalized inverse.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{N} \left\| \sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathbf{w}_i^T \cdot \mathrm{x}_j + \mathrm{b}_i\right) - \mathbf{t}_j \right\| = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \right\| \tag{11}$$

Where

$$\mathbf{H} = \begin{bmatrix} \mathrm{G}\left(\mathbf{w}_1^T \cdot \mathrm{x}_1 + \mathrm{b}_1\right) & \cdots & \mathrm{G}\left(\mathbf{w}_L^T \cdot \mathrm{x}_1 + \mathrm{b}_L\right) \\ \vdots & \ddots & \vdots \\ \mathrm{G}\left(\mathbf{w}_1^T \cdot \mathrm{x}_N + \mathrm{b}_1\right) & \cdots & \mathrm{G}\left(\mathbf{w}_L^T \cdot \mathrm{x}_N + \mathrm{b}_L\right) \end{bmatrix} \tag{12}$$

is called the hidden layer output matrix of the neural network. Equation (11) can be reformulated as $\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}$, where $\mathbf{H}^{\dagger}$ is the MP generalized inverse of $\mathbf{H}$.

3.3. Theoretical Foundation of the Proposed TNNRW

In order to deal with tensorial data directly, for a set of *N* tensor data $\left\{\left(\mathrm{X}_j, \mathbf{t}_j\right)\right\}_{j=1}^{N}$, with $\mathbf{t}_j \in \mathbf{R}^C$, Formula (9) can be reformulated as follows:

$$\mathrm{y}_j = \sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathrm{x}_j \times_1 \mathrm{u}_i^{(1)^T} \times_2 \mathrm{u}_i^{(2)^T} \cdots \times_M \mathrm{u}_i^{(M)^T} + \mathrm{b}_i\right) \tag{13}$$

where $\mathrm{X}_j \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_M}$ is a *M*th-order tensor, $\mathbf{u}_i^{(m)} \in \mathbf{R}^{I_m}$ are corresponding transformation vectors for $m = 1, \cdots, M$, $\mathrm{b}_i \in \mathbf{R}$, and $\boldsymbol{\beta}_i \in \mathbf{R}^C$, $i = 1, \cdots, L$ and *L* is the number of hidden nodes.

As the derivation in [17], the corresponding **w** is determined by only $\left(I_1 + I_2 + \cdots + I_M\right)$ variables through TNNRW while there is $\left(I_1 I_2 \cdots I_M\right)$ elements to be calculated for NNRW. In a word, utilizing the tensor based conversion in Equation (13), the input tensor data sets can be calculated directly without vectoring them. Not only the inner structural information among the elements of the data can be preserved, but also fewer parameters need to be computed. All the weights and bias will be determined randomly. After confirming the projecting vectors and biases, the output weights $\boldsymbol{\beta}$ can be determined by solving Equation (10) while the matrix **H** should be changed as follows

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathrm{G}\left(\mathrm{X}_1 \prod_{m=1}^{M} \times_m \mathbf{u}_1^{(m)^T} + \mathrm{b}_1\right) & \cdots & \mathrm{G}\left(\mathrm{X}_1 \prod_{m=1}^{M} \times_m \mathbf{u}_L^{(m)^T} + \mathrm{b}_L\right) \\ \vdots & \ddots & \vdots \\ \mathrm{G}\left(\mathrm{X}_N \prod_{m=1}^{M} \times_m \mathbf{u}_1^{(m)^T} + \mathrm{b}_1\right) & \cdots & \mathrm{G}\left(\mathrm{X}_N \prod_{m=1}^{M} \times_m \mathbf{u}_L^{(m)^T} + \mathrm{b}_L\right) \end{bmatrix} \tag{1}$$

$\mathbf{\Phi}$ is the hidden output matrix for TNNRW, then the output weights can be solved by the following optimal:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{N} \left\| \sum_{i=1}^{L} \boldsymbol{\beta}_i g\left(\mathbf{X}_j \prod_{m=1}^{M} \times_m \mathbf{u}_i^{(m)^T} + b_i \right) - \mathbf{t}_j \right\| = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{\Phi}\boldsymbol{\beta} - \mathbf{T} \right\| = \mathbf{\Phi}^{\dagger}\mathbf{T} \tag{15}$$

Where

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1^L \\ \vdots \\ \boldsymbol{\beta}_L^L \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} \mathbf{t}_1^L \\ \vdots \\ \mathbf{t}_L^L \end{pmatrix} \tag{2}$$

With the above description, this algorithm is called tensorial neural networks with random weights for tensorial data classification.

Applying the proposed TNNRW method for face recognition, for a new tensor input $\mathcal{X} \in \mathbb{R}^{m \times n \times Q}$, where $m \times n$ is the dimension of each face image, and Q is the amount of the input images. The output of the proposed TNNRW can be computed in Equation (13), and the obtained $\mathbf{y}_j$ should be compared with the real corresponding class label value $\mathbf{t}_j$ to check the effectiveness of the learning algorithm.

4. Performance Evaluation

In this section, the proposed solution of MPCA plus TNNRW is evaluated in two gait databases. The University of South Florida (USF) [18] gait database that has 1870 gait sequences from 122 individuals is the largest publicly available human gait database. These gait sequences were captured under different conditions (shoe types, walking surfaces and viewing angles). Another database is Gallery set that contains 725 sequences (subjects) and seven experiments (probe sets) are designed for human identification [19]. There are 731 samples in the Gallery set and an average of roughly 10 samples available in each subject. MPCA is used for feature extraction, TNNRW is used for classification, and the correct classification rate (CCR) is used for preliminary testing [20-21].

The first test is conducted in USF database. The first four samples from each subject (284 in total) are used for training and the rest 447 samples are used for testing. With the same feature extraction method MPCA, Figure 2 shows the CCRs comparison of three different classifiers. The three-order tensor features can be input directly for the proposed TNNRW and NN while for SVM, they need to be vectorized first. In this test, we use the first *P* numbers of extracted features from each mode of the training samples to calculate the performance of each classifier and the scale of *P* is from 1 to 10. Furthermore, both of TNNRW and NNRW set the number of hidden nodes with 1000. As seen from Figure 2, that MPCA+TNNRW achieves the highest accuracies in all cases and the advantage is obvious. This is benefit by the reservation of the inner structure of the original data for TNNRW and the classification superiority for NNRW.

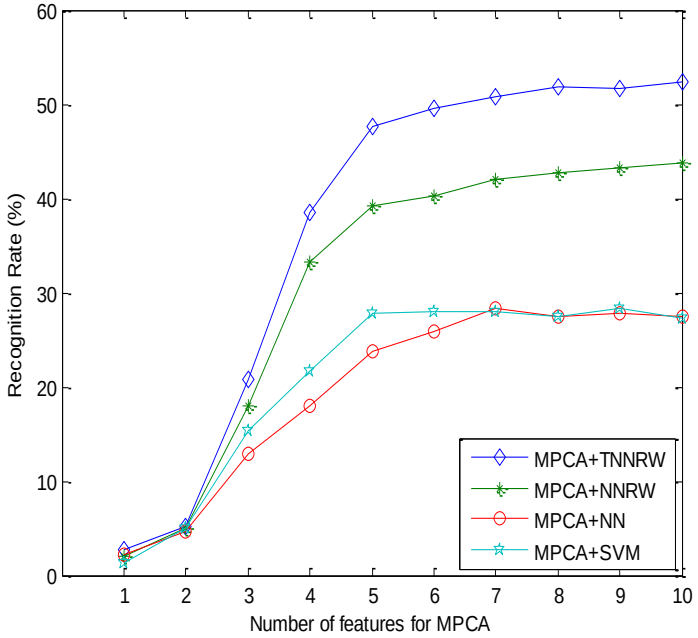

Figure 2. Recognition Rate Comparison of the Classifiers with MPCA as the Feature Extraction Tool

In the second experiment, the first five samples of each sequences (355 in total) from the gallery set are used for training and the rest used as the test data. The result captured with the MPCA+TNNRW on the training samples and test samples are shown in Figure 3. From the Figure 3, MPCA+TNNRW algorithm provides the best recognition accuracy among all other algorithms.

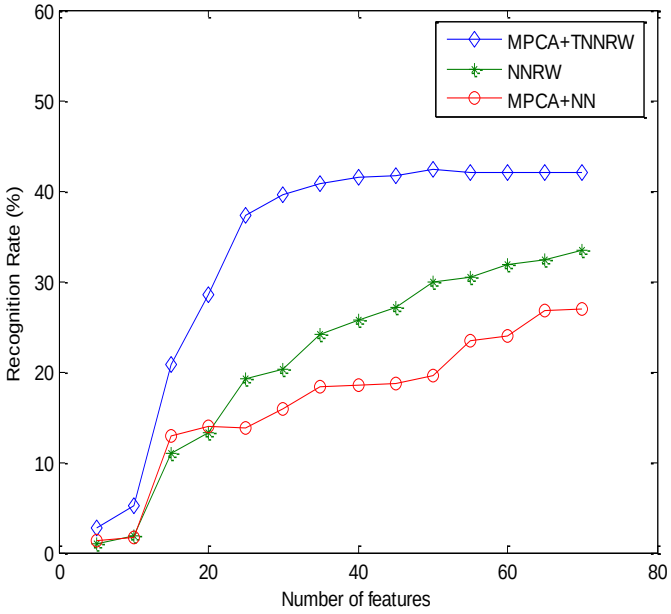

Figure 3. The Recognition Rate over Dimensions of Feature

5. Conclusions

In this paper, a new TNNRW classification is designed to classify tensorial data directly for gait recognition. For the proposed algorithm, instead of using the high dimensional input weight vector in hidden layer, a set of projecting vectors with random values are applied to preserve the natural structure information of the input sample. In addition, MPCA is used to extract features to reduce the dimension of the input data. There are two advantages of using TNNRW for gait classification, one is higher recognition rate can be obtained by classifying tensorial data directly, the other is faster computing speed can be achieved benefits by much less parameters to be calculated with TNNRW than the traditional NNRW.

Acknowledgements

The research of this paper is sponsored jointly by the program for innovative research team in Science and Technology in Universities of Henan Province under Grant No. 13IRTSTHN023, National Major Science and Technology Project under Grant No. 2014ZX03005001, Zhengzhou Science and Technology Research Plan under Grant no. 153PKJGG113, and Guangxi Natural Science Foundation Project under Grant no. 2014GXNSFCA118014.

References

[1] N. V. Boulgouris, D. Hatzinako and K. N. Plataniotis, "Gait recognition: a challenging signal processing technology for biometric identification", IEEE Signal Processing Magazine, vol. 22, no. 6, **(2005)**, pp. 78-90.
[2] Y. Huang, D. Xu and T. J. Cham, "Face and human gait recognition using image-to-class distance", IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 3, **(2010)**, pp. 431-438.
[3] H. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Boosting LDA with regularization on MPCA features for gait recognition", Biometrics Symposium, **(2007)**, pp. 1-6.
[4] M. Turk and A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, **(1991)**, pp. 71-86.
[5] P. N. Belhumeur, J. P. Hespanha and D. J. Krigman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, **(1997)**, pp. 711-720.
[6] H. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects", IEEE Transactions on Neural Networks, vol. 19, no. 1, **(2008)**, pp. 18-39.
[7] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang and H. J. Zhang, "Multilinear discriminant analysis for face recognition", IEEE Transactions on Image Processing, vol. 16, no. 1, **(2007)**, pp. 212-220.
[8] M. J. Er, S. Wu, J. Lu and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks", IEEE Transactions on Neural Networks, vol. 13, no. 3, **(2002)**, pp. 697-710.
[9] J. Qin and Z. S. He, "A SVM face recognition method based on Gabor-featured key points", Proceeding 4th IEEE Conference on Machine Learning and Cybernetics, **(2005)**, pp. 5144-5149.
[10] H. Lu, K. Plataniotis and A. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data", Pattern Recognition, vol. 44, no. 7, **(2011)**, pp. 1540-1551.
[11] J. Lu, J. Zhao and F. Cao, "Extended feed forward neural networks with random weights for face recognition," Neurocomputing, vol. 136, **(2014)**, pp. 96-102.
[12] I. Tyukin I and D. Prokhorov, "Feasibility of random basis function approximators for modeling and control," arXiv: 0905.0677, **(2009)**.
[13] L. De Lathauwer, B. De Moor and J. Vandewalle, "On the best rank-1 and rank-($R_1, R_2, \cdots, R_N$) approximation of higher-order tensors," SIAM Journal on Matrix Analysis and Applications, vol. 21, no. 4, **(2000)**, pp. 1324-1342.
[14] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications", SIAM review, vol. 51, no. 3, **(2009)**, pp. 455-500.
[15] G. B. Huang, H. Zhou, X. Ding and R. Zhang, "Extreme learning machine for regression and multiclass classification", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42, no. 2, **(2012)**, pp. 513-529

[16] W. F. Schmidt, M. Kraaijveld and R. P. W. Duin, "Feed forward neural networks with random weights," Pattern Recognition Conference B: Pattern Recognition Methodology and Systems, **(1992)**, pp. 1-4.
[17] F. Zhang, L. Qi and E. Chen, "Extended extreme learning machine for biometric signal classification", Journal of Computational and Theoretical Nanoscience, vol. 12, no. 7, **(2015)**, pp. 1247-1251.
[18] S. Sarkar, P. J. Phillips and Z. Liu, "The humanid gait challenge problem: Data sets, performance, and analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, **(2005)**, pp. 162-177.
[19] N. V. Boulgouris, K. N. Plataniotis and D. Hatzinakos, "Gait recognition using linear time normalization", Pattern Recognition, vol. 39, no. 5, **(2006)**, pp. 969-979.
[20] Q. Zhong, J. Zhao and C. Tong, "Design and Implement of Entertainment and Competition Humanoid ", Review of Computer Engineering Studies, vol. 2, no. 1, **(2015)**, pp. 17-24.
[21] T. Ling, "Super-resolution Reconstruction Method Integrated with Image Registration", Review of Computer Engineering Studies, vol. 2, no. 1, **(2015)**, pp. 31-34.

Authors

Fan Zhang, She received the MSc. in Communication and Information System from Nanjing University of Science and Technology, P. R. China, in 2006. Zhang is currently working toward the PhD degree in Zhengzhou University, P. R. China. Since 2006, she has been a Lecturer with the School of information engineering in North China University of Water Resources and Electric Power, Zheng zhou, P. R. China. Her research interests cover pattern recognition and image processing.

