Optica Applicata, Vol. XLIV, No. 2, 2014
DOI: 10.5277/oa140209

Fast near-infrared palmprint recognition using nonnegative matrix factorization extreme learning machine

XU XUEBIN[1], ZHANG XINMAN[1], LU LONGBIN[1], DENG WANYU[1], ZUO KUNLONG[2]

[1]MOE Key Lab for Intelligent Networks and Network Security,
School of Electronics and Information Engineering, Xi'an Jiaotong University,
Xi'an, 710049, China

[2]Huawei Central Research Academy, Beijing, 100095, China

[*]Corresponding author: zhangxinman@mail.xjtu.edu.cn

Support vector machine and artificial neural network are widely used in classification applications. Extreme learning machine (ELM) is a novel and efficient learning algorithm based on the generalized single hidden layer feed forward networks, which performs well in classification applications. The research results have shown the superiority of ELM with the existing classical algorithms: support vector machine (SVM) and back propagation neural network. In this study, we firstly propose a novel nonnegative matrix factorization extreme learning machine (NMFELM) to improve the performance of standard ELM method. Then we propose a novel near-infrared palmprint recognition approach based on NMFELM classifier. As the test data, we use the near-infrared palmprint database provided by Hong Kong Polytechnic University. The experimental results demonstrate that the proposed NMFELM method outperforms the standard ELM- and SVM-based methods.

Keywords: extreme learning machine, palmprint recognition, superior speed, support vector machine.

1. Introduction

Recently, a near-infrared (NIR) palmprint recognition technique has received a lot of attention and interest from the researchers [1, 2]. The NIR signature of the images can provide more discriminative information to improve the recognition accuracy [3, 4]. Usually, feature extraction and classification are very important in the NIR palmprint recognition system.

A large number of feature extraction techniques have been proposed. Principal component analysis (PCA) [5], 2D principal component analysis (2DPCA) [6], nonnegative matrix factorization (NMF) [7, 8], local linear embedding [9], local Fisher discriminant analysis [10], unsupervised discriminant projection [11], linear discriminant projection [12], sparsity preserving discriminant analysis [13], and marginal Fisher analysis (MFA) [14] are well-known feature extraction techniques, and have been widely used in many applications such as face recognition and palmprint recognition, *etc*.

On the other hand, a lot of methods have been developed for classification. The most representative methods include the classical Bayesian decision theory, artificial neural network, and traditional support vector machine (SVM) [15], *etc*. In the past decades, tremendous successes have been obtained in different research fields by SVM-based methods [15–17].

Recently, a simple and efficient learning algorithm for single-hidden layer feed forward neural networks (SLFNs), called extreme learning machine (ELM), has been proposed by GUANG-BIN HUANG *et al*. [18, 19]. ELM has been successfully applied to a number of real-world applications [19], showing a good generalization performance with extremely fast learning speed. In ELM, input weights and biases can be randomly assigned and output weights can be analytically determined by a simple generalized inverse operation [18, 19]. Compared with SVM-based methods [20–23], ELM not only learns much faster with higher generalization ability but also avoids many difficulties, such as the stopping criteria, parameter setting, learning epochs, learning rate and local minima. Due to these advantages, ELM has been successfully applied to many classification problems and has received many researchers' interest [19]. A new human face recognition algorithm based on the standard ELM and multidimensional PCA has been proposed in 2011 by MOHAMMED *et al*. [20]. QI YUAN *et al*. have proposed a novel EEG classification method based on ELM [23].

The standard ELM classifier overcomes many issues in traditional learning algorithms. However, standard ELM has some disadvantages:

– The performance of ELM is unstable. The generalization performance of the ELM algorithm depends on the proper selection of the parameters (for example, input weights) [22], especially for fewer number of training samples. It is very difficult to find the best parameters such that the training and testing accuracy is maximum for a given problem.

– For high dimensional dataset such as image data, ELM needs more hidden neurons than back propagation (BP). Thus it is important to address the existing problem of an ELM.

To address the problem of ELM, we propose a novel learning machine called a nonnegative matrix factorization extreme learning machine (NMFELM) in this paper. It is a one-stage learning machine with dimension reduction. Then, a NIR palmprint recognition method based on the nonnegative projection ELM is proposed in this paper.

The objective of this study is to improve the performance of the ELM: training and classification speed, stability, and classification accuracy. We conducted this evaluation on the NIR palmprint database provided by Hong Kong Polytechnic University. With this study, we intended to demonstrate that the proposed method outperforms the SVM-based methods and standard ELM-based methods.

2. NIR palmprint recognition based on NMFELM

In this section, we first review the basic concept of ELM. Then we propose a novel NIR palmprint recognition method based on NMFELM.

2.1. Extreme learning machine

We review the basic concept of ELM and its algorithm in this section [18, 22, 23].

Firstly, we define some symbols as follows: n – number of samples, k – label of samples, $\tilde{N}$ – number of hidden neurons, m – dimensions of dataset, $\mathbf{w}_{(m+1)\times\tilde{N}}$ – input weights matrix, $\mathbf{H}_{n\times\tilde{N}}$ – output matrix of hidden layer, $\mathbf{P}_{n\times(m+1)}$ – input in matrix format, $\mathbf{T}_{n\times 1}$ – target in matrix format, $\mathbf{Z}_{n\times\tilde{N}}$ – intermediate matrix into hidden layer, and

$$\mathbf{H}_{n\times\tilde{N}} = \begin{bmatrix} g(z_{11}) & \dots & g(z_{1\tilde{N}}) \\ \dots & \dots & \dots \\ g(z_{n1}) & \dots & g(z_{n\tilde{N}}) \end{bmatrix}_{n\times\tilde{N}}$$

$$\mathbf{P}_{n\times(m+1)} = \left\{ \left[\mathbf{x}_k, 1\right] \middle| \mathbf{x}_k \in \mathbf{R}^m \right\}_{k=1}^{n}$$

$$\mathbf{T} = \left\{ \mathbf{t}_k \middle| \mathbf{t}_k \in \mathbf{R}^l \right\}_{k=1}^{n}$$

$$\mathbf{Z} = \begin{bmatrix} z_{11} & \dots & z_{1\tilde{N}} \\ \dots & \dots & \dots \\ z_{n1} & \dots & z_{n\tilde{N}} \end{bmatrix}_{n\times\tilde{N}} = \begin{bmatrix} \mathbf{w}_1\cdot[\mathbf{x}_1, 1] & \dots & \mathbf{w}_{\tilde{N}}\cdot[\mathbf{x}_1, 1] \\ \dots & \dots & \dots \\ \mathbf{w}_1\cdot[\mathbf{x}_n, 1] & \dots & \mathbf{w}_{\tilde{N}}\cdot[\mathbf{x}_n, 1] \end{bmatrix}_{n\times\tilde{N}}$$

For n samples $\{(\mathbf{x}_k, \mathbf{t}_k)\}_{k=1}^{n}$, where $\mathbf{x}_k = [x_{k1}, x_{k2}, ..., x_{km}]$ and $\mathbf{t}_k = [t_{k1}, t_{k2}, ..., t_{km}]$, a standard SLFN with $\tilde{N}$ hidden neurons and activation function g is mathematically modeled as

$$\mathbf{o}_k = \sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g\left(\mathbf{w}_i, [\mathbf{x}_k, 1]\right), \qquad k = 1, ..., n$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{im}, w_{i(m+1)}]$ is the weight vector connecting the i-th hidden neuron with the input neurons; $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{il}]$ is the weight vector connecting the i-th hidden neuron and the output neurons; $\mathbf{o}_k = [o_{k1}, o_{k2}, ..., o_{kl}]$ is the k-th output vector of the SLFN, and $w_{i(m+1)}$ is the bias of the i-th hidden neuron. Set $\mathbf{w}_i \cdot [\mathbf{x}_k, 1]$ be the inner product of $\mathbf{w}_i$ and $[\mathbf{x}_k, 1]$. These n equations can be written compactly as

$$\mathbf{O} = \mathbf{H}\boldsymbol{\beta}$$

where

$$\mathbf{O} = \begin{bmatrix} \mathbf{o}_1 \\ \cdots \\ \mathbf{o}_n \end{bmatrix}_{n \times m}, \quad \mathbf{H} = \begin{bmatrix} g(z_{11}) & \cdots & g(z_{1\tilde{N}}) \\ \cdots & \cdots & \cdots \\ g(z_{n1}) & \cdots & g(z_{n\tilde{N}}) \end{bmatrix}_{n \times \tilde{N}}, \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \cdots \\ \boldsymbol{\beta}_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times l}$$

and

$$\mathbf{Z} = \begin{bmatrix} z_{11} & \cdots & z_{1\tilde{N}} \\ \cdots & \cdots & \cdots \\ z_{n1} & \cdots & z_{n\tilde{N}} \end{bmatrix}_{n \times \tilde{N}} = \begin{bmatrix} \mathbf{w}_1 \cdot [\mathbf{x}_1, 1] & \cdots & \mathbf{w}_{\tilde{N}} \cdot [\mathbf{x}_1, 1] \\ \cdots & \cdots & \cdots \\ \mathbf{w}_1 \cdot [\mathbf{x}_n, 1] & \cdots & \mathbf{w}_{\tilde{N}} \cdot [\mathbf{x}_n, 1] \end{bmatrix}_{n \times \tilde{N}}$$

In order to train an SLFN, one may wish to find some specific $\hat{\mathbf{w}}_i$, $\hat{\boldsymbol{\beta}}_i$ $(i = 1, ..., \tilde{N})$ such that $\|\mathbf{H}(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{\tilde{N}})\hat{\boldsymbol{\beta}} - \mathbf{T}\| = \min_{w_i, \beta} \|\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}})\boldsymbol{\beta} - \mathbf{T}\|$ which is equivalent to minimizing the cost function

$$E = \sum_{j=1}^{n} \left[\sum_{i=1}^{\tilde{N}} \beta_i g\left(\mathbf{w}_i, [\mathbf{x}_k, 1]\right) - \mathbf{t}_j \right]^2$$

where $\mathbf{H}$ are the unknown gradient-based learning algorithms which are generally used to search the minimum of $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|$.

GUANG-BIN HUANG *et al.* proposed an SLFN training algorithm called an extreme learning machine (ELM) [18]. As rigorously proved by GUANG-BIN HUANG *et al.* [18], ELM can work as a universal approximator: it is not difficult to find that SLFNs with at most n hidden neurons and with almost any nonlinear activation function can exactly learn n distinct observations. So, unlike traditional gradient-based learning algorithms, input weights of an SLFN can be randomly chosen (according to any continuous sampling distribution), and the output weights of an SLFN can be analytically determined by Moore–Penrose generalized pseudo-inverse. ELM can be summarized simply as follows [18].

ELM algorithm. Given a training set $\aleph = \{(\mathbf{x}_k, \mathbf{t}_k) | \mathbf{x}_k \in \mathbf{R}^n, \mathbf{t}_k \in \mathbf{R}^m, k = 1, 2, ..., N\}$, an activation function $g(x)$, and the number of hidden neurons $\tilde{N}$.

1) Randomly assign the input weights according to some continuous probability density function.

2) Calculate the hidden layer output matrix $\mathbf{H}$.

3) Calculate the output weight $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$ by using a simple least square method.

2.2. Nonnegative matrix factorization extreme learning machine

Extreme learning machine (ELM) randomly assigns the input weights according to some continuous probability density function. It will cause the output weights $\boldsymbol{\beta}$ different in each calculating. Thus the recognition rate in biometric applications is unstable. We propose a novel learning algorithm called NMFELM in this subsection. NMFELM needs a fewer hidden neurons and the recognition rate is stable in biometric applications.

NMF has been shown to be a very useful technique in approximating high dimensional data where the data comprise nonnegative components [7, 8, 24]. The NMF problem can be stated in a generic form as follows.

NMF problem. Given a nonnegative matrix $\mathbf{X} \in \mathbf{R}^{m \times n}$ and a positive integer $r \leq \min(m, n)$. We need to find two nonnegative matrices $\mathbf{U}$ and $\mathbf{V}$ to minimize the functional f

$$f(\mathbf{U}, \mathbf{V}) := \frac{\|\mathbf{X} - \mathbf{U}\mathbf{V}\|^2}{2}$$

The product $\mathbf{UV}$ is called a NMF of $\mathbf{X}$, although $\mathbf{X}$ is not necessarily equal to the product $\mathbf{UV}$. Clearly the product $\mathbf{UV}$ is of rank at most r. An appropriate decision on the value of r is critical in practice, but the choice of r is very often problem dependent. The objective function $f(\mathbf{U}, \mathbf{V})$ can be modified in different ways to reflect the application need. The general NMF algorithm is detailed as follows.

NMF algorithm. Calculate $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{X} \approx \mathbf{UV}$. Given a matrix with size $m{\times}n$, and $0 < r \leq \min(m, n)$:

1) $U_{i,j}$ – nonnegative values $(1 \leq i \leq m,\ 1 \leq j \leq r)$, $V_{i,j}$ – nonnegative values $(1 \leq i \leq r,\ 1 \leq j \leq n)$;
2) Scale columns of $\mathbf{U}$ to sum to one;
3) Repeat until converge or stop:

$$V_{c,j} \Leftarrow V_{c,j} \frac{\mathbf{U}^{\mathrm{T}}\mathbf{X}}{(\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{V})_{cj} + \varepsilon}, \qquad (1 \leq c \leq r,\ 1 \leq j \leq n)$$

$$U_{i,c} \Leftarrow U_{i,c} \frac{\mathbf{X}\mathbf{V}^{\mathrm{T}}}{(\mathbf{U}\mathbf{V}\mathbf{V}^{\mathrm{T}})_{ic} + \varepsilon}, \qquad (1 \leq i \leq m,\ 1 \leq c \leq k)$$

The main procedure of the NMFELM is as follows in detail.

Proposed NMFELM algorithm. Given a training set $\aleph = \{(\mathbf{x}_k, \mathbf{t}_k) | \mathbf{x}_k \in \mathbf{R}^n,\ \mathbf{t}_k \in \mathbf{R}^m,\ k = 1, 2, ..., N\}$, an activation function $g(x)$.

1) Let $\mathbf{P} = \left\{[\mathbf{x}_k, 1] \,|\, \mathbf{x}_k \in \mathbf{R}^m\right\}_{k=1}^{n}$, and $\mathbf{T} = \left\{\mathbf{t}_k \,|\, \mathbf{t}_k \in \mathbf{R}^l\right\}_{k=1}^{n}$;
2) Set the number of hidden neurons $\tilde{N}$;
3) Calculate NMF of $\mathbf{P}$: NMF($\mathbf{P}$) = [$\mathbf{U}$, $\mathbf{V}$] where $r = \tilde{N}$;

4) Set the input weights. Let $\mathbf{w} = \mathbf{U}$.

5) Calculate the hidden layer output matrix $\mathbf{H}$.

6) Calculate the output weights $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

7) Input the testing dataset. Then we can calculate the hidden layer output matrix $\mathbf{H}$ directly.

8) Calculate the output matrix $\mathbf{O} = \mathbf{H}\boldsymbol{\beta}$ directly. The classification results will be obtained.

In our NMFELM method, only output weights need be calculated by a simple least square method, the input weights have been known after NMF. Thus the recognition rate of NMFELM is stable in biometric applications. Additionally, its learning speed and classification speed are very fast. Obviously, it can be used to build a real-time NIR palmprint recognition system.

2.3. NIR palmprint recognition using NMFELM

Figure 1 displays the procedure of the near-infrared palmprint recognition algorithm via the nonnegative matrix factorization extreme learning machine.

Firstly, we extract the region of interest (ROI) of NIR palm images by using the ROI location method in [25, 26]. The gray level of all ROI images should be scaled to [0, 1], and the size should be normalized to $m \times n$ pixels.Then, we use training samples to train a NMFELM classifier. The training procedure has been introduced in Section 2.2. At last, all testing samples are applied to the NMFELM classifier to determine the NIR palmprint classification.

Figure 2 shows the difference between the one-stage learning method (our NMFELM method) and the two-stage learning method (NMF+ELM). Traditional machine learning techniques are not effective when dealing with high dimensional data. A natural idea is projecting data in high dimension space to low dimension space by dimension reduction methods, then training the classifiers on the preprocessed data. We call this method as the two-stage learning method. Its disadvantage is that the system has to store original data and preprocessed data simultaneously which will increase the space and computation complexity. Additionally, this will make it difficult for on-line learning and on-line updating. In fact, this two-stage method is somewhat redundant in dimension reduction and training neural network.

One-stage learning method can address the problem, by removing redundancy and combining dimension reduction with neural network training. Through dimension reduction, we not only reduce the dimension but also obtain the number of hidden neurons and input weights of SLFN simultaneously. This size-fixed neural network will become linear programming system and thus the output weights can be determined by a simple least square method. Unlike the ELM proposed by Huang that assigns input weights randomly, and becomes unstable for high dimensional data, NMFELM's input weights are assigned by a fixed nonnegative value. Thus the recognition rate of NMFELM is stable. The experimental results will show the superiority of NMFELM method over ELM-based methods.

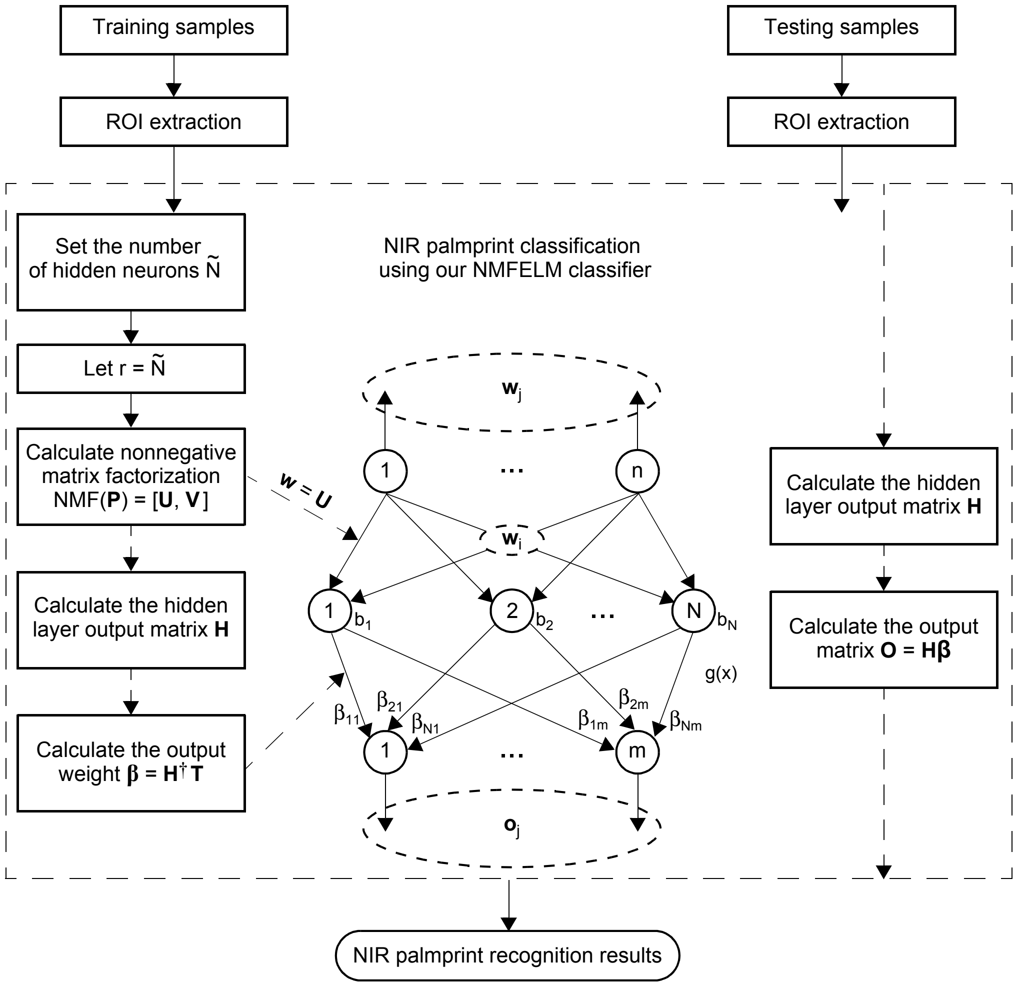


Fig. 1. Near-infrared palmprint recognition based on the proposed NMFELM method.

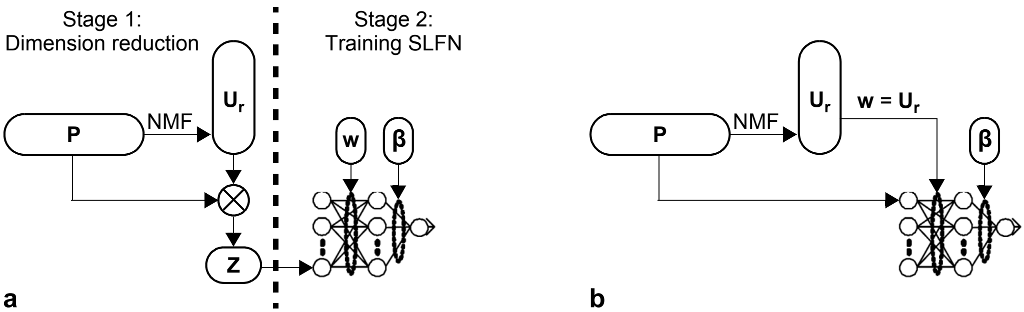


Fig. 2. The difference between two stage learning procedure (NMF+ELM) – **a**, and one-stage learning procedure (NMFELM) – **b**.

3. Data preparation and experimental design

3.1. Experimental database

In our experiments, we use the multispectral palmprint database provided by Hong Kong Polytechnic University [1, 2]. The multispectral database contains images captured with near-infrared and visible light. The images were collected from 250 volunteers, including 55 females and 195 males. The age distribution is from 20 to 60 years old. It has a total of 6000 images obtained from 500 different palms. These images were collected in two stages. The time interval between the first and the second stage was about 9 days. In each stage, the volunteer was asked to provide 6 images for each palm. Therefore, 12 images of each illumination were collected from each palm (left and right palm).

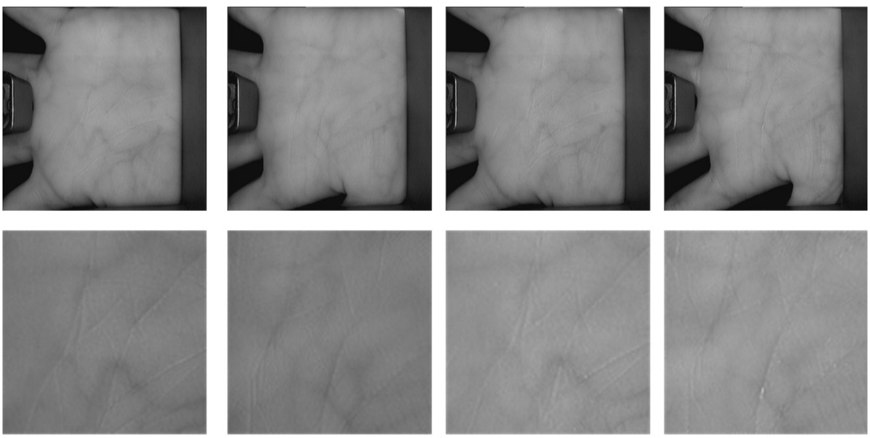
3.2. Training and testing samples

In the experiments, we use 6000 NIR palmprint images from 500 palms. Each palm has 12 images which are used in our experiments. All ROI images are normalized to $m \times n$ pixels in our experiments. The image samples are shown in Fig. 3.

In the experiment, we randomly choose 3 NIR images of each palm for training. The rest 9 NIR images are considered as the testing samples. In order to evaluate the performance of our method, we have 30 times random selection of the training dataset and take the average recognition rate as the final experimental results. So 1500 NIR palmprint images are used as training samples, and 4500 NIR palmprint images are used as testing samples in the experiment.

3.3. Experimental parameters setting

All ROI images are normalized to $m \times n$ pixels. In our experiments, $m = n =$ = [32, 40, 45, 50]. The NMFELM, ELM and classical SVM classifiers are used in our



a

b

Fig. 3. Image samples. Original NIR palmprint images (**a**) and ROI images (**b**).

experiments. The activation functions $g(x)$ = sigmoid(x) and $g(x) = \sin(x)$ are used in the ELM and NMFELM classifiers. For NMFELM and ELM classifiers, the nearly optimal numbers of hidden neurons are selected based on the cross-validation method.

In our experiments, we use sequential minimal optimization support vector machine (SMOSVM) for its high performance [16, 17]. To solve multi-class classification problem, there are two most popular approaches: one-against-all method and one-against-one method. To reduce the training time and classification time, we use one-against-all method [15, 22]. So we constructed 500 binary SVM classifiers in the experiments. For SMOSVM method, Gaussian radial basis function is used as a kernel function and the parameters setting (C and γ) are selected based on the cross-validation method [15, 16, 27].

4. Experimental results and discussion

A comparative study among SVM-, ELM- and NMFELM-based methods will be presented in this section. Here MFA+ELM [14], NMF+ELM [7], MFA+SMOSVM, NMF+SMOSVM, NMFELM, competitive code [1, 2, 26], 2DPCA+ELM [28], 2DPCA+SMOSVM [28] methods are used in the experiments.

4.1. Experimental results

The experimental results using Hong Kong Polytechnic University NIR palmprint database are shown in Table 1. Three NIR images/classes were used as training samples in the experiment. In Table 2, the proposed NMFELM method achieves the best recognition accuracy of 99.48% (32×32, sigmoid), 99.31% (40×40, sin), 99.51% (45×45, sigmoid) and 99.36% (50×50, sigmoid). In addition, we can also find that the standard ELM-based methods outperform SMOSVM-based methods.

T a b l e 1. Recognition accuracy using 3 training samples.

Methods (functions)	Recognition accuracy (different image size $m \times n$)			
	32×32	40×40	45×45	50×50
NMFELM (sigmoid)	99.48	99.15	99.51	99.36
NMFELM (sin)	99.43	99.31	99.46	99.29
NMF+ELM (sigmoid)	99.22	98.82	99.26	99.15
NMF+ELM (sin)	99.20	98.78	99.24	99.12
MFA+ELM (sigmoid)	92.23	94.35	95.59	96.55
MFA+ELM (sin)	92.03	94.21	95.48	96.46
MFA+SMOSVM	92.86	93.11	95.45	95.86
NMF+SMOSVM	96.43	96.12	97.25	97.12
Competitive code	98.52	98.56	98.92	98.71
2DPCA+ELM (sigmoid)	98.12	98.01	99.17	99.13
2DPCA+ELM (sin)	98.09	97.83	99.09	99.06
2DPCA+SMOSVM	94.86	94.68	95.25	95.42

Table 2. The computational cost using 3 training samples [4].

Methods (functions)	Training time [s]			
	32×32	40×40	45×45	50×50
NMFELM (sigmoid)	4.01	5.62	7.10	8.67
NMFELM (sin)	3.96	5.93	6.98	8.23
NMF+ELM (sigmoid)	5.96	8.86	10.11	10.85
NMF+ELM (sin)	5.94	8.84	10.08	10.72
MFA+ELM (sigmoid)	2.83	4.53	6.02	8.17
MFA+ELM (sin)	2.80	4.51	5.99	8.15
MFA+SMOSVM	74.81	76.01	78.23	80.87
NMF+SMOSVM	80.72	82.12	83.11	85.12
Competitive code	–	–	–	–
2DPCA+ELM (sigmoid)	5.11	7.97	9.65	10.06
2DPCA+ELM (sin)	5.05	7.93	9.54	9.96
2DPCA+SMOSVM	76.43	78.82	80.02	82.71
Methods (functions)	Classification time [ms]			
	32×32	40×40	45×45	50×50
NMFELM (sigmoid)	0.051	0.065	0.069	0.083
NMFELM (sin)	0.049	0.063	0.065	0.081
NMF+ELM (sigmoid)	0.085	0.094	0.097	0.108
NMF+ELM (sin)	0.069	0.085	0.089	0.095
MFA+ELM (sigmoid)	0.075	0.089	0.106	0.108
MFA+ELM (sin)	0.069	0.072	0.102	0.104
MFA+SMOSVM	9.36	9.41	9.47	9.51
NMF+SMOSVM	6.47	7.04	7.11	7.33
Competitive code	–	–	–	–
2DPCA+ELM (sigmoid)	0.087	0.091	0.095	0.101
2DPCA+ELM (sin)	0.079	0.083	0.087	0.092
2DPCA+SMOSVM	7.51	8.14	8.29	8.45

In Table 1, we reported the experimental results of our method on four different image resolutions. With the increment of the image size, the recognition rate reaches the maximum (45×45). Then the recognition rate will be descended.

4.2. Computational cost

The computational cost of our method is reported in this section. Table 2 show the training time and average classification time using different methods. We use the same NIR palmprint images as Section 4.1 in the experiments. The algorithms are implemented by MATLAB 7.12 and C++ and performed on a personal computer (Intel Core i5-2430 2.8 GHz CPU, 4096 M RAM, Windows XP SP3). Each experiment is executed 100 times and the average data are reported.

We can see from Table 2 that NMFELM-based methods are very fast. For example, training time of NMF+SMOSVM (size 50×50) method is 85.12 s in Table 2, while that of NMFELM (sigmoid, size 50×50) is 8.67 s. NMFELM method is 9.8 times faster than NMF+SMOSVM method. The classification time of NMFELM (sigmoid, size 50×50) method is 0.083 ms, while that of NMF+SMOSVM is 7.33 ms. NMFELM method is 88 times faster than NMF+SMOSVM method. Both training speed and classification speed, in NMFELM method is faster than in standard ELM-based methods and SVM-based methods. The conclusion can be drawn that our NMFELM method needs less computational time and obtains higher accuracy than SMOSVM methods and standard ELM-based methods.

In addition, the total computational time of competitive code method is 135.18 s (32×32), 158.25 s (40×40), 206.43 s (45×45), 260.52 s (50×50). It is slower than SMOSVM-based method.

4.3. Stability analysis

Figure 4 shows the recognition accuracy of fix sized NMFELM, NMF+ELM over 50 trials with same training dataset, respectively. We can find that NMFELM is more stable than ELM-based method. This is because the input weights of NMFELM were

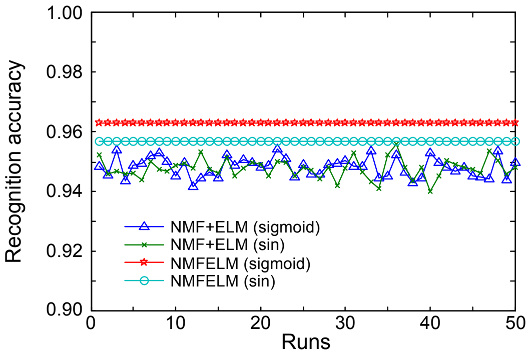


Fig. 4. Recognition accuracy of NMFELM, NMF+ELM over 50 trials with the same training dataset.

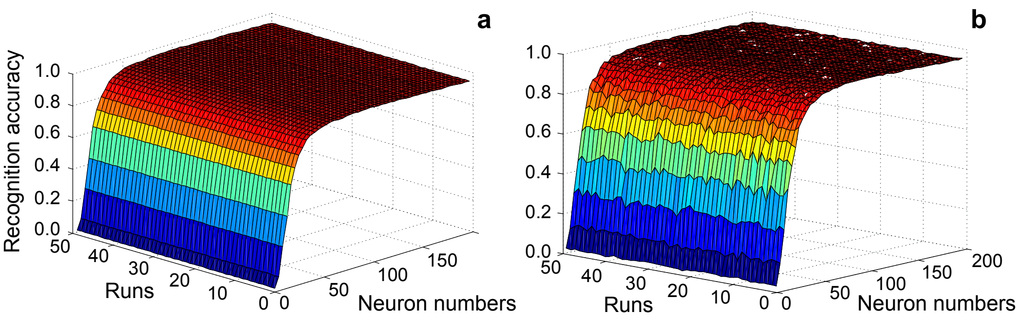


Fig. 5. Accuracy of NMFELM (**a**) and NMF+ELM (**b**) variation with respect to neuron numbers (sigmoid).

obtained from vectors of dataset by NMF, while the input weights of ELM are randomly assigned. Obviously, the performance of ELM is very sensitive to initial values of input weights. The input weights of NMFELM are fixed and nonnegative. This makes the NMFELM easy to obtain the optimal setting.

Figure 5 shows the recognition accuracy variation with respect to hidden neuron numbers. In each trial, the performance variation curve with respect to hidden neurons is entirely same, and this indicates that NMFELM avoids the influence of random input weights in ELM. We can more clearly find that NMFELM is more stable than NMF+ELM method.

5. Conclusions

In this paper, we present a near-infrared palmprint recognition method based on nonnegative matrix factorization extreme learning machine (NMFELM). The training speed and classification speed of NMFELM method are very fast. NMFELM is also more stable than ELM. Experimental results on Hong Kong Polytechnic University NIR palmprint database demonstrate its effectiveness and robustness. It is clear that NMFELM is a novel and effective classifier for near-infrared palmprint recognition, and can be used in other applications such as image classification, synthetic aperture radar (SAR) image processing, video searching, *etc*.

Acknowledgements – The work is supported by the grants from the National Natural Science Foundation of China (No. 60972146 and No. 61100166), Huawei Innovation Research Program, the Fundamental Research Funds for the Central Universities, the grant from China Postdoctoral Science Foundation (No. 20110491661) and the special financial grant from China Postdoctoral Science Foundation (No. 2012T50807).

References

[1] ZHANG D., ZHENHUA GUO, GUANGMING LU, ZHANG D., WANGMENG ZUO, *An online system of multispectral palmprint verification*, IEEE Transactions on Instrumentation and Measurement **59**(2), 2010, pp. 480–490

[2] YING HAO, ZHENAN SUN, TIENIU TAN, CHAO REN, *Multispectral palm image fusion for accurate contact-free palmprint recognition*, 15th IEEE International Conference on Image Processing, ICIP, 2008, pp. 281–284.

[3] LI S.Z., RUFENG CHU, SHENGCAI LIAO, LUN ZHANG, *Illumination invariant face recognition using near-infrared images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(4), 2007, pp. 627–639.

[4] DONG YI, RONG LIU, RUFENG CHU, RUI WANG, DONG LIU, LI S.Z., *Outdoor Face Recognition Using Enhanced Near Infrared Imaging*, Advances in Biometrics, Lecture Notes in Computer Science, Vol. 4642, 2007, pp. 415–423.

[5] YIN-NONG CHEN, CHIN-CHUAN HAN, CHENG-TZU WANG, KUO-CHIN FAN, *Face recognition using nearest feature space embedding*, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(6), 2011, pp. 1073–1086.

[6] WANGMENG ZUO, KUANQUAN WANG, ZHANG D., *Bi-directional PCA with assembled matrix distance metric*, IEEE International Conference on Image Processing, ICIP, Vol. 2, 2005, pp. 958–961.

[7] LEE D.D., SEUNG H.S., *Learning the parts of objects by non-negative matrix factorization*, Nature **401**(6755), 1999, pp. 788–791.

[8] PAUCA V.P., PIPER J., PLEMMONS R.J., *Nonnegative matrix factorization for spectral data analysis*, Linear Algebra and its Applications, **416**(1), 2006, pp. 29–47.

[9] ROWEIS S.T., SAUL L.K., *Nonlinear dimensionality reduction by locally linear embedding*, Science **290**(5500), 2000, pp. 2323–2326.

[10] SUGIYAMA M., *Local Fisher discriminant analysis for supervised dimensionality reduction*, ICML '06 Proceedings of the 23rd international conference on Machine Learning, AMC Press, 2006, pp. 905–912.

[11] JIAN YANG, ZHANG D., JING-YU YANG, NIU B., *Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(4), 2007, pp. 650–664.

[12] HONGPING CAI, MIKOLAJCZYK K., MATAS J., *Learning linear discriminant projection for dimensionality reduction of image descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(2), 2011, pp. 338–352.

[13] LISHAN QIAO, SONGCAN CHEN, XIAOYANG TAN, *Sparsity preserving discriminant analysis for single training image face recognition*, Pattern Recognition Letters **31**(5), 2010, pp. 422–429.

[14] DONG XU, SHUICHENG YAN, DACHENG TAO, LIN S., HONG-JIANG ZHANG, *Marginal Fisher analysis and its variants for human gait recognition and content-based image retrieval*, IEEE Transactions on Image Processing **16**(11), 2007, pp. 2811–2821.

[15] BIN QI, CHUNHUI ZHAO,EUNSEOG YOUN, NANSEN C., *Use of weighting algorithms to improve traditional support vector machine based classifications of reflectance data*, Optics Express **19**(27), 2011, pp. 26816–26826.

[16] LOPEZ J., DORRONSORO J.R., *Simple proof of convergence of the SMO algorithm for different SVM variants*, IEEE Transactions on Neural Networks and Learning Systems **23**(7), 2012, pp. 1142–1147.

[17] TA-WEN KUAN, JING-FA WANG, JIA-CHING WANG, PO-CHUAN LIN, GAUNG-HUI GU, *VLSI design of an SVM learning core on sequential minimal optimization algorithm*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **20**(4), 2012, pp. 673–683.

[18] GUANG-BIN HUANG, QIN-YU ZHU, CHEE-KHEONG SIEW, *Extreme learning machine: theory and applications*, Neurocomputing **70**(1–3), 2006, pp. 489–501.

[19] GUANG-BIN HUANG, DIANHUI WANG, *Advances in extreme learning machines (ELM2010)*, Neurocomputing **74**(16), 2011, pp. 2411–2412.

[20] MOHAMMED A.A., MINHAS R., WU Q.M.J., SID-AHMED M.A., *Human face recognition based on multidimensional PCA and extreme learning machine*, Pattern Recognition **44**, 2011, pp. 2588–2597.

[21] YUGUANG WANG, FEILONG CAO, YUBO YUAN, *A study on effectiveness of extreme learning machine*, Neurocomputing **74**(16), 2011, pp. 2483–2490.

[22] WAN-YU DENG, QING-HUA ZHENG, SHIGUO LIAN, LIN CHEN, XIN WANG, *Ordinal extreme learning machine*, Neurocomputing **74**(1–3), 2010, pp. 447–456.

[23] QI YUAN, WEIDONG ZHOU, SHUFANG LI, DONGMEI CAI, *Epileptic EEG classification based on extreme learning machine and nonlinear features*, Epilepsy Research **96**(1–2), 2011, pp. 29–38.

[24] PINGHUA GONG, CHANGSHUI ZHANG, *Efficient nonnegative matrix factorization via projected Newton method*, Pattern Recognition **45**(9), 2012, pp. 3557–3565.

[25] XIAO-YUAN JING, YONG-FANG YAO, DAVID ZHANG, JING-YU YANG, MIAO LI, *Face and palmprint pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition*, Pattern Recognition **40**(11), 2007, pp. 3209–3224.

[26] ZHANG D., WAI-KIN KONG, YOU J., WONG M., *Online palmprint identification*, IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(9), 2003, pp. 1041–1050.

[27] ERGUN B., KAVZOGLU T., COLKESEN I., SAHIN, C., *Data filtering with support vector machines in geometric camera calibration*, Optics Express **18**(3), 2010, pp. 1927–1936.

[28] JIAN YANG, ZHANG D., FRANGI A.F., JING-YU YANG, *Two-dimensional PCA: a new approach to appearance-based face representation and recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(1), 2004, pp. 131–137.

Received October 28, 2013
in revised form April 28, 2014