Fuzzy sparse autoencoder framework for single image per person face recognition

Guo, Yuwei; Jiao, Licheng; Wang, Shuang; Wang, Shuo; Liu, Fang

DOI:
10.1109/TCYB.2017.2739338

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Guo, Y, Jiao, L, Wang, S, Wang, S & Liu, F 2018, 'Fuzzy sparse autoencoder framework for single image per person face recognition', *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2402-2415. https://doi.org/10.1109/TCYB.2017.2739338

Link to publication on Research at Birmingham portal

Fuzzy sparse autoencoder framework for single image per person face recognition

Yuwei Guo, *Student Member, IEEE*, Licheng Jiao, *Senior Member, IEEE*, Shuang Wang, *Member, IEEE*, Shuo Wang, *Member, IEEE* and Fang Liu, *Senior Member, IEEE*

***Abstract*—The issue of single sample per person face recognition has attracted more and more attention in recent years. Patch/local based algorithm is one of the most popular categories to address the issue, as patch/local features are robust to face image variations. However, the global discriminative information is ignored in patch/local based algorithm, which is crucial to recognize the non-discriminative region of face images. To make the best of the advantage of both local information and global information, a novel two-layer local-to-global feature learning framework is proposed to address single sample per person face recognition. In the first layer, the objective-oriented local features are learnt by a patch-based fuzzy rough set feature selection strategy. The obtained local features are not only robust to the image variations, but also usable to preserve the discrimination ability of original patches. Global structural information is extracted from local features by a sparse autoencoder in the second layer, which reduces the negative effect of non-discriminative regions. Besides, the proposed framework is a shallow network, which avoids the over-fitting caused by using multi-layer network to address single sample per person problem. The experimental results have shown that the proposed local-to-global feature learning framework can achieve superior performance than other state-of-the-art feature learning algorithms for single sample per person face recognition.**

***Index Terms*—two-layer feature learning, fuzzy rough set, sparse autoencoder, one sample per person face recognition**

I. Introduction

Face recognition has received great attention in the past few years [1]–[4]. It has been applied in various areas, such as information security and smart card applications. A variety of face recognition algorithms have been proposed [5]–[9], including global feature based algorithms and local feature based algorithms. For example, two classical algorithms, principle component analysis (PCA) [10] and linear discriminant analysis (LDA) [11], belong to global feature based algorithms. The representative local feature based algorithms include local binary patterns (LBP) [12] and gabor wavelets [13].

Single sample per person (SSPP) face recognition is an active branch of face recognition. It is necessary to study the issue of SSPP as only one labeled sample per person is available in many practical applications, such as identify card identification, passport verification and law enforcement [14]. Some algorithms have been proposed to address the problem of SSPP face recognition so far. These algorithms can be generally classified into three main categories [14]: generic learning based algorithms, virtual sample generation based algorithms and patch/local based algorithms.

For the first category, a separate face dataset, which is called generic training set, is needed. The generic training set includes multiple samples per person, and these samples have possible variances in expression, illumination, etc. Discriminative features can be extracted from the generic training set to help to solve SSPP problems. For example, based on the discriminative information from the generic training set and the essential collaborative representative relationship between the gallery set and generic training set, collaborative probabilistic labels (CPL) is proposed in [15]. Typical generic learning based algorithms include adaptive generic learning (AGL) [16] and sparse representation classifier (SRC) based algorithms [17], [18]. One main assumption in generic learning based algorithms is that generic training set and the gallery set share similar information in both of the within-class variations and the between-class variations. However, the assumption is too strong in some cases. It is not easy to collect a generic training set containing various skin colors, ages or occupations in practical applications, which has a significant impact on the effectiveness of generic learning based algorithms.

For the second category, extra samples for each person are generated. The discriminant information is learned from the single sample and extra samples. For example, two singular value decomposition (SVD) based perturbation algorithms are proposed in [19] and [20] to obtain extra images for each person. Authors in [21] synthesize virtual samples by projecting an image with an arbitrary expression into the expression subspaces. It is believed that images with the same expression are located on a common expression subspace. In general, prior knowledge is needed to generate new virtual images. However, it is not guaranteed the quality and reality of the generated virtual images [22]. Besides, the generated virtual samples are highly correlated among each other. Therefore, the obtained discriminative features from the virtual samples

This work was supported by the National Basic Research Program (973 Program) of China, No. 2013CB329402, the National Natural Science Foundation of China, No. 61573267 and No. 61473215, The Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project), No. B07048, the Major Research Plan of the National Natural Science Foundation of China, No. 91438201 and No. 91438103, the Program for Cheung Kong Scholars and Innovative Research Team in University, No. IRT_15R53.

Yuwei Guo, Licheng Jiao, Shuang Wang and Fang Liu are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation of China, Xidian University, Xi'an, Shaanxi Province 710071, China. (e-mail: yuweiguo18@126.com)

Shuo Wang is with the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, UK (e-mail: s.wang@cs.bham.ac.uk)

may be redundant [23], [24].

For the third category, each face image is always divided into several local patches and the discriminant learning techniques are used to extract features. For example, local spectral features are learnt to represent the face images to enlarge the training set [25]. Authors in [26] divide each face image into multiple non-overlapping local patches and extract local binary pattern (LBP) features from each patch. Liu et al. adopt the divide-conquer-aggregate strategy to address SSPP problems [27]. First, each face image is divided into local patches, and each local patch is then classified and integrated results. In [28], the local patches are considered as a manifold, and the SSPP problem is formulated as a manifold-manifold matching problem. The prominent advantage of the third category algorithm is the robustness to image variations in lighting, expression and occlusion, and easily avoiding the affection of severely corrupted non-informative regions [29]. However, they only focus on the local relationships and tend to ignore the global discriminative information of face image, which may be very important for classification. The global information is crucial to the recognition of the non-discriminative regions, such as forehead and cheek.

Deep learning has been an extremely active research area in recent years [30]–[32], which has achieved an enormous success in face recognition with a large number of labeled samples [33]–[36]. Nowadays, deep learning is tried to extend to deal with SSPP problem [37], [38]. In [37] and [38], good quality frontal images (gallery images) are referred to the only one sample per person. The faces with all type of variants (such as lighting, expression or poses) are regarded as noise images. As we know, large-scale data is the key to the success of deep learning, which is due to there are many parameters in multi-layer networks. To fully train the parameters and avoid over-fitting, massive data is necessary. Therefore, both good quality frontal images and the faces with all type of variants are used to train a deep neural network. That is, the number of training samples per person is greater than one in [37] and [38].

To address face recognition with only one training sample per person, a novel two-layer local-to-global feature learning framework (TLFL) is proposed to extract discriminative features. Different from multi-layer learning, the proposed framework is a shallow network, which avoids the over-fitting caused by using multi-layer network to address SSPP problem. Besides, as both local feature and global feature are extracted from the only one training sample per person, the proposed framework is not only robust to image variations in lighting, expression and poses, but also reduce the effect of non-discriminative regions of face images. In the first layer, a patch-based fuzzy rough set feature selection strategy is used to select objective-oriented local features. The local features preserve the discrimination ability of original patches and are robust to the image variations, however, may suffer from non-discriminative regions. To reduce the effect of non-discriminative regions, global structural information is then extracted from the collected local features by a sparse autoencoder in the second layer, which reduces the redundancy of the local features. The output of sparse autoencoder is the final obtained features for recognition. The effectiveness of the framework is tested on several face image datasets.

The main advantages of the proposed framework can be summarized as follows.

- The proposed framework integrates the advantages of both local information and global information. Besides, the framework is a shallow network, which avoids the over-fitting caused by using multi-layer network to address SSPP problem.
- Discriminant information is introduced to learn the objective-oriented local features by the patch-based fuzzy rough set feature selection strategy. The main advantages of the patch-based fuzzy rough set feature selection strategy are: (1) it does not need any preliminary or additional information about data. (2) The selected local features can fully characterize the knowledge of the original patches and remain the discrimination ability of the original patches. (3) Patch-based strategy needs a much less memory space than original image based fuzzy rough set feature selection.
- Global information is extracted from the collected local features by a sparse autoencoder, which mines the structural information and reduces the redundancy of the collected local features.

The rest of this paper is organized as follows. In Section II, the theoretical backgrounds of fuzzy rough set and sparse autoencoder are described. The two-layer feature learning framework is proposed in Section III. Experimental results and parametric analysis on face images are described in Section IV. Conclusions and feature work are given in Section V.

II. Preliminaries

In this section, we briefly review the definitions of fuzzy rough set and sparse autoencoder.

A. Basic Concepts of Fuzzy Rough Set

Given a fuzzy information system $FS = \langle U, C \cup D, V, f \rangle$, B is a subset of condition feature C, for arbitrary feature $a \in C - B$, the gain of feature $\widetilde{Gain}(a, B, D)$ can be defined as:

$$\widetilde{Gain}(a, B, D) = \widetilde{I}\left(B \cup \{a\}; D\right) - \widetilde{I}\left(B; D\right) \tag{1}$$

where, $\widetilde{I}\left(B; D\right) = \widetilde{H}\left(D\right) + \widetilde{H}\left(B\right) - \widetilde{H}\left(BD\right)$. If $B = \Phi$, $\widetilde{Gain}(a, B, D) = \widetilde{I}\left(\{a\}; D\right)$.

$\widetilde{H}\left(B\right)$ and $\widetilde{H}\left(D\right)$ are information quantity of the fuzzy indiscernibility relation, which are defined as follows:

$$H\left(\widetilde{s}\right) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_{\widetilde{s}}\right|}{n} \tag{2}$$

where, s is B or D,$[x_i]_{\widetilde{s}}$ denotes the fuzzy equivalence class generated by fuzzy indiscernibility relation $\widetilde{s}$.

$\widetilde{H}\left(BD\right)$ is defined as the joint entropy of B and D, which is defined as:

$$\widetilde{H}\left(BD\right) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_{\widetilde{B}} \cap [x_i]_{\widetilde{D}}\right|}{n} \tag{3}$$

The $\widetilde{Gain}(a, B, D)$ indicates whether the equivalence classes change when adding feature a. If there are any changes, then the feature a will be selected.

A modified formula $\widetilde{Gain\_Ratio}(a, B, D)$, which is called the mutual information gain ratio of feature a, is proposed for feature selection in [39]. $\widetilde{Gain\_Ratio}(a, B, D)$ is obtained by:

$$\widetilde{Gain\_Ratio}(a, B, D) = \frac{\widetilde{Gain}\left(a, B, D\right)}{\widetilde{H}\left(\{a\}\right)} \tag{4}$$

If $B = \Phi$, $\widetilde{Gain\_Ratio}(a, B, D) = \frac{\widetilde{I}(\{a\};D)}{\widetilde{H}(\{a\})}$. The $\widetilde{Gain\_Ratio}$ will be used in the first layer of our proposed algorithm.

B. Sparse Autoencoder

Sparse autoencoder is usually regarded as an element in deep neural networks [40], which contains three layers: input layer, hidden layer and output layer. The three layers are all fully connected as illustrated in Fig. 1. The number of units in L1 and L3 are fixed to the dimension of the input sample. Sparse autoencoder can be used to reduce features if the number of units in L2 is smaller than the number of units in L1. The purpose of using the autoencoder is to find a latent feature representation by learning a function $h_{w,b}\left(x\right) \approx x$. That is, to minimize the reconstruction error between the input samples and the output samples.

Let $X = \{x_i\}_{i=1}^{N}$ is the input samples and $Y = \{y_i\}_{i=1}^{N}$ is the output samples, the reconstruction error is defined as:

$$J\left(W, b\right) = \left[\frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{1}{2}\left\|x_i - y_i\right\|_2^2\right] + \frac{1}{\lambda}\sum\nolimits_{i,j,l}\left(W_{i,j}^{(l)}\right)^2 \tag{5}$$

where, N is the total number of samples, l is the layer number of the sparse autoencoder, $W_{i,j}^{(l)}$ represents the weight in l-th layer. The first item is the error term, which is constructed by L2-norm. The second is the regular term, which is used to prevent the over-fitting. λ denotes attenuation parameter weight, which controls the relative importance of these two terms.

In order to achieve the sparseness of the hidden units, an extra penalty factor is added [41]. It keeps the average activity of hidden units within a small range, which is denoted as:

$$KL(\rho||\widehat{\rho}_j) = \rho\log\frac{\rho}{\widehat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho}_j} \tag{6}$$

where, ρ is the parameter representing sparseness, which measures the target average activity; $\widehat{\rho}_j$ shows the average activity of j unit over the input data. $KL(\rho||\widehat{\rho}_j)$ represents the relative entropy between two Bernoulli random variables: one is ρ, and the other is $\widehat{\rho}_j$. The finally cost function is defined as:

$$J_{sparse}\left(W, b\right) = J\left(W, b\right) + \beta\sum_{j=1}^{s}KL\left(\rho||\widehat{\rho}_j\right) \tag{7}$$

where, β is the weight controlling sparsity penalty factor; s is the number of units in hidden layer.

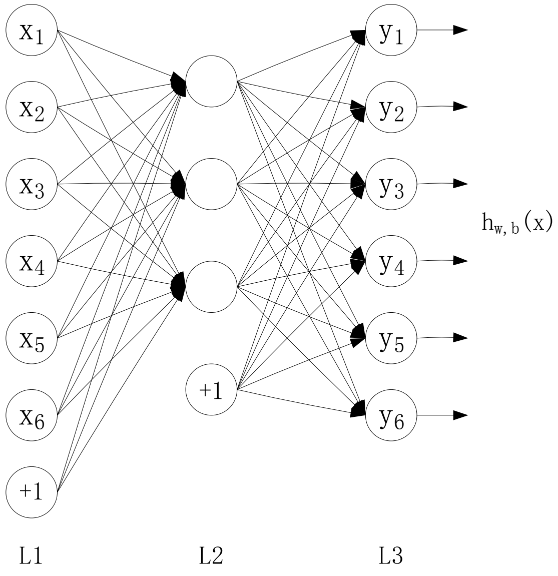

Fig. 1. Framework of sparse autoencoder.

III. Two-layer local to global feature learning framework

In this section, a two-layer local-to-global feature learning framework (TLFL) is proposed for single image per person face recognition. First layer: objective-oriented local features are extracted by the patch-based fuzzy rough set feature selection strategy. Second layer: Sparse autoencoder is used to extract the global structural information from the collected local features. TLFL is illustrated in Fig. 2. Feature 1 are objective-oriented local features. Feature 2 are global features extracted from Feature 1 by a sparse autoencoder. The details of TLFL framework is shown in the following subsection.

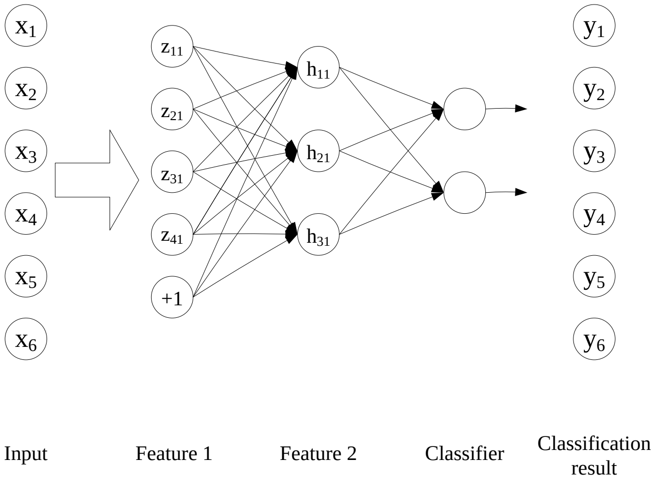

Fig. 2. The simple framework of TLFL.

A. First Layer: Local Features Extracted by Patch-based Feature Selection Strategy

Patch/local based feature learning algorithms are one of the main categories to address SSPP problem [25], [26].

These algorithms always partition the original image into non-overlapping patches with a fixed size and feature extraction is performed in each isolated patch. However, first, the local feature extraction considers the structure of image but ignores the discriminant information. Second, the relevant information between neighboring pixels that are located in two patches is not evaluated [42].

To obtain objective-oriented local feature and make full use of the relevant information between neighboring pixels, Fuzzy rough set is used to select features from each overlapping patch, which is named PRS. The main advantages of PRS is: First, it does not need any preliminary or additional information about data [43]. Second,local features is obtained according to the relevance between the labels and features, which can fully characterize the knowledge of the original data and remain the discrimination ability of the original data. Third, patch-based fuzzy rough set strategy requires a much less memory space than original image based fuzzy rough set strategy. For example, if the original image is a 26×26 matrix, then the fuzzy indiscernibility relation matrix in fuzzy rough set is a 676×676 matrix. However, if a patch-based fuzzy rough set strategy is used and the patch size is 3×3, the relation matrix of a patch is only a 9×9 matrix. The relation matrix of all patches require a much less memory space ($9 \times 9 \times 676$) than that of the original image.

Overlapping patches is obtained by expanding individual pixel to a patch. The patch of a pixel consists of the pixel and its adjacent pixels. The patch size is an odd number. Note that, when the patch size is 1, the patch of a pixel is the pixel itself. Suppose that N input training images $\{T_i\}_{i=1}^{N}$ are given, and the size of each image is $m \times n$. For the i-th image, around each pixel, we take a $k_1 \times k_2$ patch, and all overlapping patches are collected. Note that, k_1 and k_2 are odd numbers. Each pixel is represented by the corresponding patch. That is, each image is composed of $m \times n$ patches. i.e., image T_i contains $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,mn}\} \in R^{k_1 k_2}$ patches, where each $x_{i,j}, j \in [1, mn]$ is the jth patch. Here, mn is short for $m \times n$. For example, suppose the size of a face image is 5×6, which is shown in Fig. 3(a). Each circle is a pixel, and there are 30 pixels. For any pixel, such as pixel 15, we take a 3×3 patch, i.e. $k_1 = 3, k_2 = 3$. The patch of pixel 15 is comprised of the pixel 15 and its neighbourhood pixels 8, 9, 10, 14, 16, 20, 21 and 22. The patch of the pixel 15 is represented by the dotted box in Fig. 3(b). To avoid the special case of the area of the patch locating outside the original face image, we have to expand the image. The size of the expanded image is $(m+k_1-1) \times (n+k_2-1)$. The up and down of the image are increased $(k_1-1)/2$ rows, respectively. The right and left of the image are increased $(k_2-1)/2$ columns, respectively. For example, the expanded image size is $(5+3-1) \times (6+3-1)$, which is shown in Fig. 3(c). The first step of expanding the image is to expand rows by copying the first row and last row, respectively. The pixels 1 to 6, which is framed by the solid box in Fig. 3(c), is the increased row for the up of image. Then, the first column on the left is copied to increase the left of the image. The increased pixels are 1, 1, 7, 13, 19, 25 and 25, which are framed by the solid box in Fig. 3(c). To increase the right of the image by copying the right-most column. Based on the expanded image, we can obtain the patch of the pixel 30, which is shown in Fig. 3(c), framed by the dotted box. The $\overline{X}_i = \{\overline{x}_{i,1}, \overline{x}_{i,2}, ..., \overline{x}_{i,mn}\}$ is achieved through subtracting patch mean from each patch, where $\overline{x}_{i,j}$ denotes a mean-removed patch. Do the above operation for all images, and we can get the re-constructed patch face images: $\overline{X} = \{\overline{X}_1, \overline{X}_2, ..., \overline{X}_N\} \in R^{k_1 k_2 \times N_{mn}}$.

In the fuzzy rough set theory, reduction can be considered as a feature subset selection process. The selected subset does not lose any discernibility of the original data [44]. Traditional reduction algorithms based on rough set theory treat each pixel as a original feature, and the feature selection procedure operates on all pixels. In PRS, a pixel in the face image is represented with a patch, and feature selection applies to each patch.

In order to better understand the PRS strategy, its framework is described in Fig. 4. As shown in Fea 1 in Fig. 4, there are $m \times n$ patches in the given face image, $Pat = \{p_1, p_2, ...p_{mn}\}$, mn is short for $m \times n$, which is represented by squares. Each patch consists of $k_1 \times k_2$ features, and the feature is shown in circles. A pixel in face image is represented by a patch, i.e. $k_1 \times k_2$ features. Feature redundancy may exist in a patch. That is, not all features in the patch are good for the recognition of the patch. In order to express a patch well, we expect to delete redundancy features from patch. The features in patch are reduced through fuzzy rough set theory. Feature selection process is performed independently in each patch. In order to select appropriate features for each patch, the relationship between features in patch and labels are considered. The main process for selecting features in each patch is shown in Algorithm 1, in which the mutual information gain ratio $Gain\_Ratio$ is employed. Different patches are represented by different amount of features. The reduced result is shown in Fea 2 of Fig. 4. Fea 2 has $m \times n$ patches at the same with the Fea 1, however, the number of the features in each corresponding patches may be different. As the result of reduction, the number of features in the patches of Fea 2 ranges from 1 to $k_1 \times k_2$. The all selected features in Fea 2 are shown in Fea 3 of Fig. 4. Fea 3 is the selected feature via the PRS strategy in the first layer, which can be reduced again by sparse autoencoder in the second layer. The number of Fea 3 is range from $m \times n$ to $m \times n \times k_1 \times k_2$.

B. Second Layer: Global Features Extracted from the Collected Local Features

The obtained local features in the first layer may suffer from the non-discriminative regions. To reduce the effect of non-discriminative regions, global features are extracted from the collected local features. Considering the rules generated by fuzzy rough set may be unstable and sparse autoencoder can be used as the postprocessing of rough set [45]. A sparse autoencoder is used to extract global information, which exploits the underlying structure of the collected local features. Suppose a sparse autoencoder with m neurons in the hidden layer and s neurons in the output layer. The input data $x \in R^n$ is the output from the PRS strategy:

$$z^{(2)} = W_1^T x + b_1, W_1 \in R^{m \times n}, b_1 \in R^m \qquad (8)$$

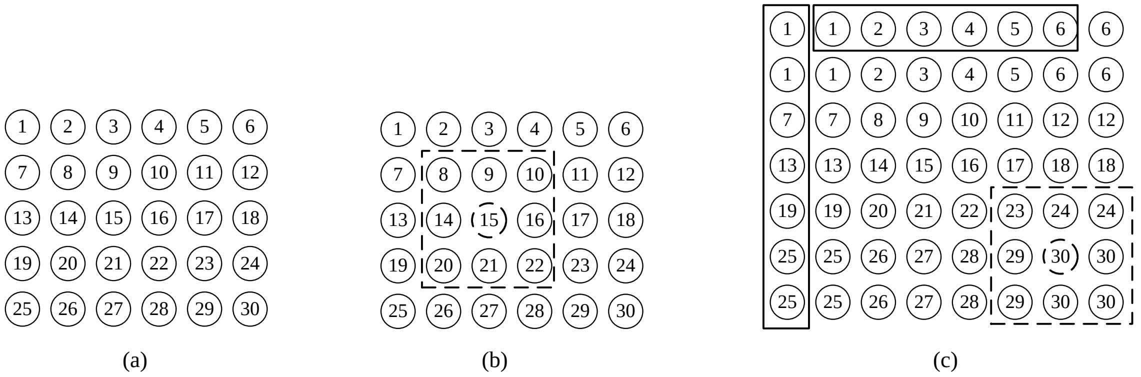

Fig. 3. The example of the patch production.

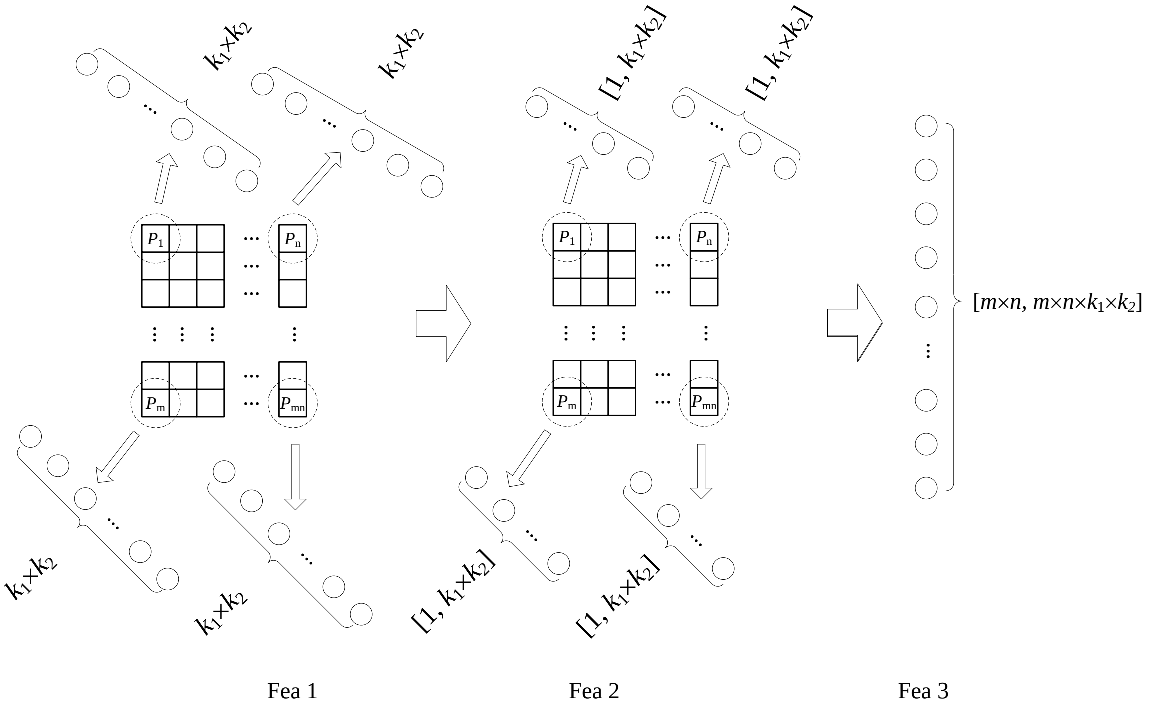

Fig. 4. The framework of PRS strategy.

Algorithm 1: Patch-based fuzzy rough set feature selection

```
Input: $U, P, D$
Output: $B = \{B_1, B_2, ...B_{mn}\}$
Initialize $B_i = \Phi, B_i \in B$
Step1:  For each patch $P_i \in P$ do
Step2:      For each feature $a \in P_i$, compute the
            significance of feature $a$, $Gain\_Ratio(a, B_i, D)$.
Step3:      Select the feature $a_m$ such that
            $G_m = \max_{a \in P_i} Gain\_Ratio(a, B_i, D)$
Step4:      $B_i \leftarrow B_i \cup \{a_\mathrm{m}\}$
Step5:  until $Gain\_Ratio(a_m, B_i, D) \leq 0$
Step6:  The set $B = \{B_1, B_2, ...B_{mn}\}$ is the selected
        feature, $B_i$ is the selected feature for $i$th patch.
```

$$a^{(2)} = f\left(z^{(2)}\right) \tag{9}$$

$$z^{(3)} = W_2^T a^{(2)} + b_2, W_2 \in R^{s \times m}, b_2 \in R^s \tag{10}$$

$$a^{(3)} = f\left(z^{(3)}\right) \tag{11}$$

where, W_i is i-th connectivity weight, b_i is the $(i+1)$-th bias term, $i = 1, 2$, $a^{(3)}$ is the output result, $f(\bullet)$ is the activated function, the commonly used is sigmoid function, $f(x) = {}^{1}\!/\!_{(1+\exp(x))}\cdot$

When we define the output value is equal to the input value, i.e. $a^{(3)} = x$, the network is called autoencoder network, which mimic the mapping from the input to itself. $a^{(2)}$ can completely describe the input value. Furthermore, if the number of units in hidden layer less than the amount of units in input layer ($m < n$), then the sparse representation of original data can be obtained. The network is the sparse autoencoder network. The penalty function is defined in Eq. 12.

$$\begin{aligned} J\left(W,b\right) = & \left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left|x-a^{(3)}\right|^{2}\right] + \frac{1}{\lambda}\sum_{i,j,l}\left(W_{i,j}^{(l)}\right)^{2} \\ & + \sum_{j=1}^{m}\left(\rho\log\left(\frac{\rho}{\rho_j}\right) + (1-\rho)\log\frac{(1-\rho)}{(1-\rho_j)}\right) \end{aligned} \qquad (12)$$

The first term in Eq. 12 is the error term, which shows the error between the input data and the output data. For ease of calculation, the error term is constructed with L2-norm. The second term is regular term, which prevents the over-fitting. The third term shows the sparse constraint term. $W_{i,j}^{(l)}$ is the weight of the i-th unit in l-th layer to the j-th unit in $(l+1)$-th layer. ρ is the sparse parameter. $\widehat{\rho_j}$ represents the average activity of j unit. Back propagation algorithm is used to compute the partial derivation of the penalty function. Given the penalty function and the partial derivation, the optimization parameters (i.e. W and b) of the network can be compute with L-BFGS algorithm. Suppose that N input images, $\left\{\overline{T_i}\right\}_{i=1}^{N}$, $\overline{T_i}$ denotes i-th input image which is treated with the PRS strategy. According to the optimized parameters, the extracted features $a^{(2)}$ are computed from Eq. 8 and Eq. 9. Finally, classifier is used to verify the performance of the two-layer reduced features. Note that, fine tuning strategy is employed to further optimize parameters.

IV. Experimental results and analysis

In order to evaluate the effectiveness of the proposed algorithm, in this section, six face image datasets are discussed. In our experiments, Extended Yale B [46], AR [47], five subsets selected from CMU PIE [48], LFW [49], CAS-PEAL [50], and JAFFE [51] face datasets are used.

A. Data Descriptions and Experimental Settings

The first face dataset is Extended Yale B, which has 2404 front-view face images. The images are collected from 38 individuals, and each individual has about 64 images under different laboratory-controlled lighting conditions.

The second dataset, AR, contains over 4000 frontal face images of 126 individuals. The images are collected under different expressions and lighting conditions. As in [52]–[54], a subset of the AR dataset, 50 male and 50 female, are selected in the experiments. There are 14 images per person, and the total number of the used images is 1400.

The third dataset is the CMU PIE face dataset, which consists of 41368 images of 68 individuals. 13 different pose, 43 different illumination and 4 different facial expression are included in the dataset. Note that, CMU PIE dataset has less illumination variations than the Extended Yale B dataset. As in [53], [54], the subset in CMU PIE are selected. We choose the subsets P05, P07, P09, P27 and P29 in the experiments. The five subsets present five different poses: looking right, up, down, front and left, respectively. In P05 and P27 dataset, each individual has about 49 images, and there are about 24 images in P07, P09 and P29 datasets.

The fourth dataset is LFW face dataset, which contains more than 13000 face images. As in [55], the subjects having more than 9 images per person are selected. There are a total of 158 subjects. The images are colored image, which are transformed into gray image.

The fifth dataset is CAS-PEAL, which contains 30863 images of 1040 individuals. The dataset is divided a training set, a gallery set and six frontal probe sets. The six set are expression set, lighting set, accessories set, background set, distance set and ageing set, respectively. The expression set is used, which contains 284 individuals.

The sixth dataset is JAFFE, which is composed of 213 images corresponding to 10 different subjects. Each subject is represented with 7 categories of expression, i.e. angry, disgust, fear, neutral, sadness, happiness and surprise.

All the face images are manually aligned and cropped to be 26 × 26 in size in the experiments. Fig. 5 shows some example face images from the ten datasets, and the five images corresponding to any dataset belong to the same person.

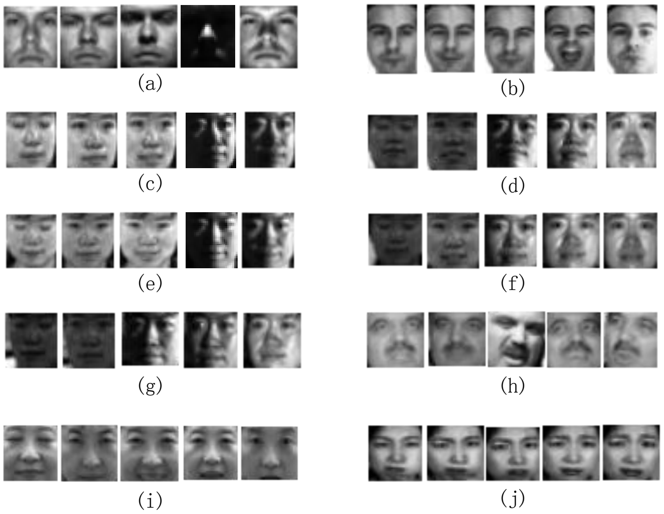
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 5. Example face images. (a) Extended Yale B dataset. (b) AR dataset. (c) P05 dataset. (d) P07 dataset. (e) P09 dataset. (f) P27 dataset. (g) P29 dataset. (h) LFW dataset. (i) CAS-PEAL dataset. (j) JAFFE dataset.

Each face dataset is split into two groups in our experiments: the testing dataset and the backup training dataset. The proportion of testing set is 70 %, and the percentage for the backup training dataset is 30 %. The training set has only one labeled sample for each person, which are selected from the backup training set. It is clear that selecting different classifiers may lead to various results of the proposed algorithm. Thus, two different classifiers are selected, i.e. softmax and linear support vector machine (linear SVM), to verify our algorithm. Sparse autoencoder is usually used together with softmax [40], [56], therefore, softmax is selected as classifier. SVM is a popular classifier in the field of machine learning [57]. As the simple predictive function and powerful generalization, a linear version of SVM is used at the classification stage. Each experiment is performed 20 times on every dataset. The average classification accuracy is recorded for comparison.

TABLE I
THE AVERAGE ACCURACY RATE AND STANDARD DEVIATION (STD) ACORSS 20 TESTS WITH ONE LABELED SAMPLE PER FACE IMAGE, USING SOFTMAX. THE BEST RESULT FOR EACH DATASET IS IN BOLD. (MEAN%±STD%)

Dataset	OriFea	AE	KFRS	GFRS	PRS
Y	23.08±2.7	21.49±2.7	22.98±2.4	23.17±2.1	**30.95±2.6**
A	36.75±3.4	32.65±2.5	34.50±3.1	35.75±3.0	**50.00±2.8**
P05	43.53±1.3	43.13±2.4	46.45±2.4	45.39±1.8	**55.29±1.6**
P07	41.30±3.5	42.13±2.4	39.26±3.2	40.81±3.1	**44.54±2.2**
P09	42.06±3.2	43.82±2.8	38.96±2.7	40.98±3.8	**48.23±1.6**
P27	43.07±2.9	44.84±1.7	42.45±2.6	42.48±2.7	**53.98±2.9**
P29	38.24±2.2	42.06±1.7	39.21±2.3	38.14±2.5	**42.98±2.4**
lfw	6.33±1.0	6.11±1.3	6.32±1.5	6.45±1.8	**13.46±1.4**
Cas	59.89±2.1	61.28±2.3	60.46±2.7	60.96±2.6	**65.43±2.5**
Jaf	78.42±4.3	80.24±4.7	79.65±4.9	79.65±4.1	**87.58±4.2**
Average	41.27	41.78	41.02	41.38	**49.24**

TABLE II
THE AVERAGE ACCURACY RATE AND STANDARD DEVIATION (STD) ACORSS 20 TESTS WITH ONE LABELED SAMPLE PER FACE IMAGE, USING SVM. THE BEST RESULT FOR EACH DATASET IS IN BOLD. (MEAN%±STD%)

Dataset	OriFea	AE	KFRS	GFRS	PRS
Y	26.37±2.7	23.90±2.3	26.79±2.1	27.53±1.9	**33.68±2.0**
A	42.00±2.7	27.25±2.6	40.65±2.2	41.35±2.7	**55.40±3.1**
P05	55.29±1.2	49.12±2.1	53.96±2.3	54.66±1.9	**58.24±2.0**
P07	48.38±1.7	45.13±1.8	45.78±2.2	46.66±1.4	**53.96±2.1**
P09	45.59±3.2	43.82±1.6	45.21±2.1	45.05±3.8	**56.24±2.9**
P27	54.28±1.3	45.13±1.4	54.96±2.3	54.23±1.9	**58.29±1.8**
P29	45.29±1.5	42.06±1.6	45.29±1.8	46.37±2.0	**47.87±2.2**
lfw	6.69±1.4	4.31±1.5	7.03±1.4	7.11±1.4	**14.33±1.5**
Cas	63.03±2.7	53.24±2.8	63.65±2.1	62.31±2.4	**70.04±2.5**
Jaf	80.53±6.3	78.95±6.6	80.64±6.0	81.37±6.5	**86.43±5.9**
Average	46.75	41.29	46.4	46.66	**53.45**

B. Impact of the Patch Size

The first experiment is to investigate the impact of the patch size on the feature selection strategy PRS in the first layer, and to decide the choice of the patch size. The patch size needs an odd number, which is set to 1, 3, 5, 7 and 9 respectively. Note that, when the patch size is 1, the patch of a pixel is the pixel itself. The impact is assessed by comparing the classification accuracy of PRS with different patch size. The accuracy curves of PRS with softmax and SVM classifier are respectively shown in Fig. 6 and Fig. 7. The x-axis is the patch size, and the y-axis is classification accuracy. It is quite clear from the curves that the performance of the PRS is influenced by the patch size. The accuracy of PRS has good performance when the patch size is 3. And the accuracy decreases when the patch size is greater than 3. Therefore, we set the size of patch to 3 in the following experiment.

C. Comparing PRS with Other Methods

PRS is a patch-based fuzzy rough set feature selection strategy. The patch size is set to 3. Table I and Table II list the classification accuracy of PRS compared to four other feature representation methods on the ten datasets using softmax and SVM, respectively. OriFea is the baseline, in which all features are used for classification. AE represents sparse autoencoder. Both KFRS and GFRS are feature reduction algorithms based on fuzzy rough set. KFRS is proposed by Hu [58]. GFRS is short for GAIN_AS_FRS, which is proposed by Dai [39]. From Table I and Table II we can see that:
(1) AE has not well performance than OriFea with classifier SVM, while AE has the similar performance with OriFea when classifier is softmax. The choice of classifier has an impact on the performance of algorithm. (2) It is difficult to get a good effect when fuzzy rough set theory is used to face recognition directly. i.e., the accuracy of PRS is superior to the accuracy of KFRS and GFRS. (3) As shown in Table I, PRS compared with other algorithm (OriFea, AE, KFRS and GFRS), the different of average accuracy with softmax is 7.97%, 7.46%, 8.22% and 7.86%, respectively. When classifier is SVM, the corresponding difference of average classification accuracy is 6.70%, 12.16%, 7.05% and 6.79%, respectively.

TABLE III
THE CLASSIFICATION ACCURACY OF DIFFERENT PARAMETER s WITH SOFTMAX.(%)

Dataset	0.1	0.3	0.5	0.7	0.9
Y	**31.27**	30.49	29.63	29.61	28.22
A	**51.00**	50.94	49.86	47.62	47.42
P05	**57.06**	56.76	55.59	56.18	55.88
P07	**45.13**	43.95	44.25	43.36	43.95
P09	**49.12**	47.35	47.94	47.06	45.88
P27	54.28	**54.57**	**54.57**	53.39	52.80
P29	42.94	**43.53**	42.94	42.94	42.65
lfw	13.21	**14.52**	12.04	11.39	11.62
Cas	**66.13**	64.95	63.27	62.98	61.76
Jaf	87.26	**88.26**	87.43	86.93	85.42

D. The Impact of the Number of Features in the Second Layer

The second layer of the proposed algorithm is to extract structural features through the sparse autoencoder. We defined parameter s is the number of final features to the number of features obtained from PRS strategy ratio, i.e.,

$$s = \frac{the \quad number \quad of \quad final \quad feature}{the \quad number \quad of \quad feature \quad after \quad processed \quad by \quad PRS \quad strategy},$$

which ranges from 0.1 to 0.9 and is of interval value 0.2.

The impact of the parameter s is discussed in two ways: for single dataset with tabular form, i.e. Table III and Table IV and for the average value on all datasets with curve form, i.e. Fig. 8. Table III and Table IV show the impact of parameter s when using classifiers softmax and SVM, respectively. The best result for each dataset is shown in boldface. From Table III and Table IV, we can see that: when the parameter s is smaller than 0.5, a better classification result can be obtained; for softmax is used as classifier, when parameter s is 0.1, TLFL get best results in most datasets. The average classification accuracy on all datasets are shown in Fig. 8, from which the following points can be seen: With the increase of parameter s, the accuracy curve decreases gradually; No matter which classifier is selected, 0.1 to 0.3 for parameter s usually lead to the best result.

E. Comparing TLFL with other algorithms

Comparison results among our proposed algorithm TLFL and other feature learning algorithms are shown in Table V

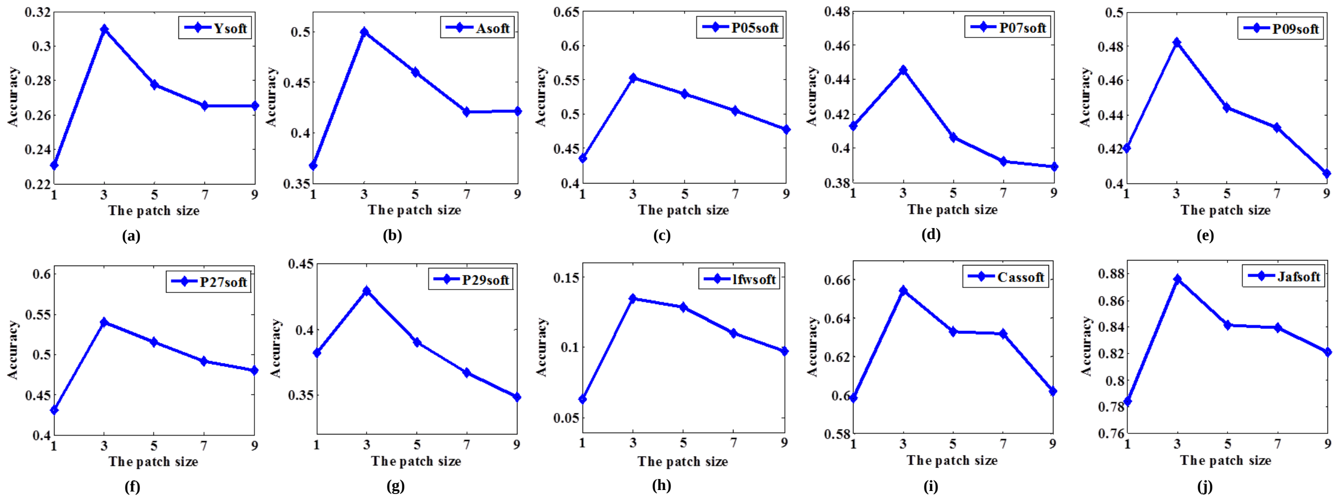

Fig. 6. The impact of patch size on PRS with softmax. (a) Extended Yale B dataset. (b) AR dataset. (c) P05 dataset. (d) P07 dataset. (e) P09 dataset. (f) P27 dataset. (g) P29 dataset. (h) LFW dataset. (i) CAS-PEAL dataset. (j) JAFFE dataset.

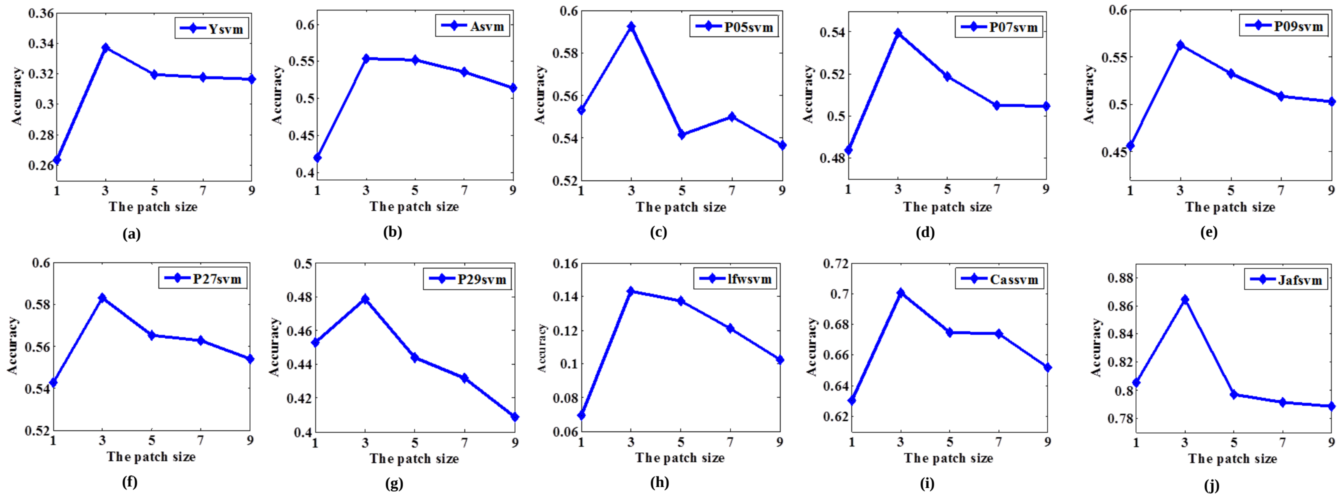

Fig. 7. The impact of patch size on PRS with SVM. (a) Extended Yale B dataset. (b) AR dataset. (c) P05 dataset. (d) P07 dataset. (e) P09 dataset. (f) P27 dataset. (g) P29 dataset. (h) LFW dataset. (i) CAS-PEAL dataset. (j) JAFFE dataset.

TABLE IV
THE CLASSIFICATION ACCURACY OF DIFFERENT PARAMETER s WITH SVM.(%)

Dataset	0.1	0.3	0.5	0.7	0.9
Y	**34.86**	32.84	31.46	30.96	30.24
A	54.32	**55.50**	55.48	53.46	52.73
P05	58.82	**59.71**	58.82	58.82	57.94
P07	**54.87**	52.51	51.62	51.03	52.21
P09	55.00	**56.47**	55.88	55.00	54.71
P27	57.52	**58.70**	57.52	57.52	57.52
P29	**49.71**	49.41	49.41	49.41	49.41
lfw	14.23	**15.21**	13.08	12.98	12.51
Cas	**72.95**	71.43	70.65	70.65	69.42
Jaf	88.20	**89.47**	87.78	87.21	86.65

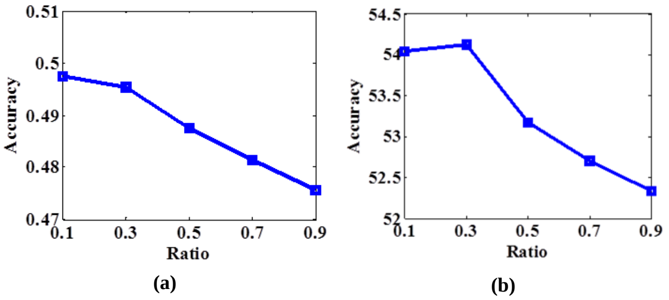

Fig. 8. Average accuracy on all datasets for each ratio. (a) classifier is softmax. (b) classifier is SVM.

and Table VI. In Table V, softmax is used as the classifier. SVM is used in Table VI. The compared algorithms include:

- OriFea means original feature are employed to train classifier.
- AE is sparse autoencoder.
- StackAE is the stacked autoencoders. For a fair comparison, two layers stacked autoencoders are chosed. Note that, AE is a special case of StackAE, which is the StackAE with one layer.

- LBP is local binary pattern, which is proposed in [12].
- HOG [59] is short for histograms of oriented gradients.
- TT is a face recognition algorithm, which is proposed by Tan [60].
- LRA [53] is used to deal with single sample based face recognition.
- DLR is proposed by Yin [54] for single labeled image per person face recognition.
- LSF-PC [25] extracts local spectral features for one sample per person problem.
- DPC is proposed in [26], which extract LBP features from non-overlapping patches.
- TLFL is our proposed algorithm.

Table V and Table VI show the comparison results with softmax and SVM classifier, respectively, from which the following conclusions can be drawn.

(1) StackAE do not act well when only one labeled training sample per face image is available. Comparison StackAE with AE, the two-layer AE has no advantage than one layer AE for one sample per person face recognition.

(2) TLFL achieves a very good classification result no matter which the classifier is used (softmax or SVM in this paper). TLFL is compared with other algorithms (OriFea, AE, StackAE, LBP, HOG, TT, LRA, DLR, LSF-PC and DPC), the different of accuracy with softmax is 8.8%, 8.39%, 11%, 27.73%, 14.59%, 3.7%, 2.34%, 1.84%, 2.38% and 1.74%, respectively. When classifier is SVM, the corresponding difference of classification accuracy is 7.98%, 13.46%, 19.28%, 31.28%, 17%, 4.45%, 2.97%, 2.3%, 3.14%, and 2.42%, respectively.

The performance of the proposed algorithm is evaluated by Cohen's kappa coefficient,which is shown in Table VII. From the comparison of the kappa coefficient, we can see that our proposed algorithm have better agreement than other algorithms. The confusion matrices of the proposed algorithm and other algorithms on JAFFE dataset are shown in Fig.9. The confusion matrices of TLFL on various datasests are expressed in the form of map, which are shwon in Fig. 10.

F. Comparing TLFL with CNN

To compare the performance of CNN model with the proposed TLFL, CNN with two convolutional layers is used to extract features from single sample per person. The code of CNN is publicly available, which is come from DeepLearnToolbox [61]. Two classifiers, softmax and SVM, are used to evaluate the performance of CNN and TLFL. We discuss the impact created by five parameters on CNN. The five parameters are shown as follows.

- $batS$ represents the proportion of samples to be updated in each iterative of stochastic gradient descent. The value is set to [0.5,1].
- $alpha$ is the declining rate in each iteration. The range of the parameter is 0.01 to 2.
- OM denotes the number of feature maps, which is set to 1 to 12.
- KS represents the size of convolutional kernel. The value is set to [1,3,5,7,9,11].

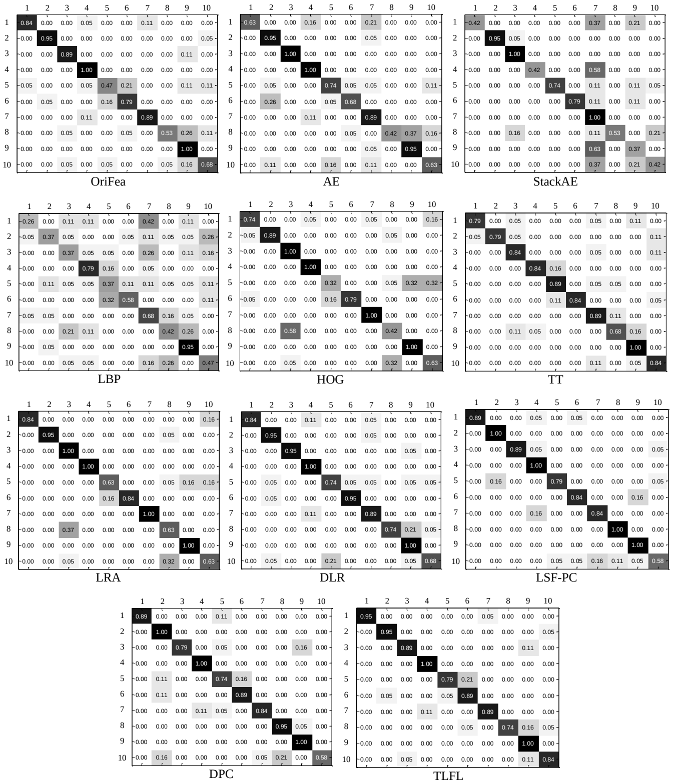

Fig. 9. Confusion matrices of various algorithms with SVM on JAFFE.

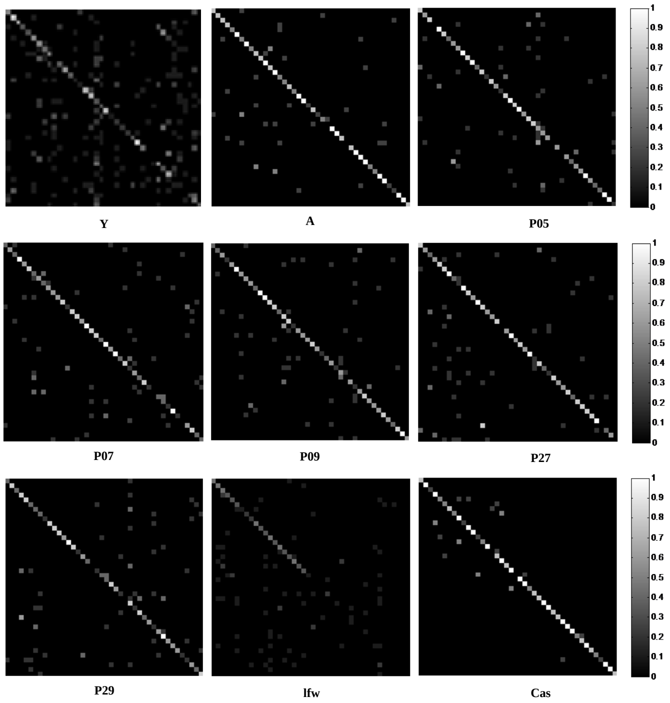

Fig. 10. Confusion matrices of TLFL on various datasets.

TABLE V

COMPARISON OF CLASSIFICATION ACCURACIES OF DIFFERENT ALGORITHMS WITH SOFTMAX. THE BEST RESULT FOR EACH DATASET IS IN BOLD. (MEAN%±STD%)

Dataset	OriFea	AE	StackAE	LBP	HOG	TT	LRA	DLR	LSF-PC	DPC	TLFL
Y	23.08±2.1	21.09±2.3	21.43±1.9	12.91±1.8	10.42±2.0	26.39±2.6	29.43±2.7	28.65±1.9	28.96±2.3	29.42±2.3	**31.27±1.9**
A	36.75±2.9	33.50±2.3	28.50±2.0	15.75±2.4	37.50±2.5	46.90±3.2	47.00±2.6	47.75±2.3	48.75±2.4	48.50±2.2	**51.00±2.5**
P05	43.53±1.3	43.13±2.4	39.41±2.1	20.00±2.3	36.43±1.7	54.39±2.1	54.27±3.1	55.98±1.3	53.64±2.8	55.15±2.8	**57.06±3.2**
P07	41.30±3.5	42.13±2.4	39.53±2.6	18.58±2.6	25.87±2.4	42.93±2.3	42.66±3.0	43.27±2.1	43.65±2.7	44.00±2.5	**45.13±2.8**
P09	42.06±3.2	43.82±2.8	38.82±2.9	19.12±3.1	30.28±3.2	46.03±2.7	46.98±2.7	47.00±1.8	46.14±2.3	47.92±2.2	**49.12±3.8**
P27	43.07±2.9	44.84±1.7	43.07±2.8	21.83±2.9	37.10±3.0	50.16±1.9	53.14±2.9	52.96±1.8	52.81±2.7	53.10±2.9	**54.57±3.3**
P29	38.24±2.2	42.06±1.7	37.94±1.9	19.71±2.8	29.95±2.6	40.11±2.1	41.63±2.1	42.93±1.8	41.52±2.4	41.77±2.4	**43.53±2.7**
lfw	6.33±1.0	6.11±1.1	5.43±1.1	3.87±1.4	8.06±1.9	10.15±1.3	12.32±1.4	13.64±1.5	12.74±2.1	13.26±1.8	**14.52±1.4**
Cas	59.86±2.8	60.36±3.0	59.00±2.7	40.49±2.6	62.63±2.2	61.59±2.2	63.46±2.8	63.96±2.6	63.15±2.4	63.96±2.9	**66.13±2.7**
Jaf	78.42±5.9	79.65±5.8	77.43±5.9	51.05±5.5	76.44±5.9	84.91±5.7	86.27±5.6	86.03±5.8	85.46±5.6	86.11±5.5	**88.26±5.8**
Average	41.26	41.67	39.06	22.33	35.47	46.36	47.72	48.22	47.68	48.32	**50.06**

TABLE VI

COMPARISON OF CLASSIFICATION ACCURACIES OF DIFFERENT ALGORITHMS WITH SVM. THE BEST RESULT FOR EACH DATASET IS IN BOLD. (MEAN%±STD%)

Dataset	OriFea	AE	StackAE	LBP	HOG	TT	LRA	DLR	LSF-PC	DPC	TLFL
Y	26.37±2.7	23.90±1.5	23.61±1.8	14.29±2.3	12.64±2.8	31.06±1.9	32.85±2.4	32.57±2.3	33.19±1.9	34.30±1.8	**34.86±1.6**
A	42.00±2.8	27.25±1.9	20.05±1.9	18.50±2.7	39.50±2.2	50.65±3.1	52.30±2.5	52.75±2.8	52.25±2.4	53.05±1.9	**55.50±2.4**
P05	55.29±1.2	49.12±2.1	39.71±2.1	22.06±3.0	37.94±2.9	56.05±2.1	56.66±2.7	57.84±2.3	54.33±2.7	56.37±2.5	**59.71±1.7**
P07	48.38±1.7	45.13±1.8	38.05±3.3	20.94±2.7	27.43±3.0	51.35±2.5	52.36±2.4	52.93±2.9	50.26±3.1	53.61±2.9	**54.87±2.9**
P09	45.59±3.2	43.82±1.6	38.24±2.0	20.88±1.9	31.47±2.1	52.46±3.4	54.21±2.3	54.24±3.1	53.37±2.8	53.14±2.5	**56.47±3.3**
P27	54.28±1.3	45.13±1.4	40.12±2.4	24.78±2.1	38.94±2.9	53.55±2.9	56.25±2.7	55.88±2.8	56.96±2.6	56.37±2.7	**58.70±2.7**
P29	45.29±1.5	42.06±1.6	37.65±2.1	21.18±2.5	31.18±2.1	45.63±3.1	46.47±2.6	48.87±2.7	47.24±3.0	46.00±2.8	**49.71±1.6**
lfw	6.96±1.4	4.31±1.7	3.28±1.6	3.48±1.9	7.59±2.1	11.69±1.6	12.43±1.5	12.10±1.4	13.60±1.7	14.25±1.7	**15.21±1.5**
Cas	63.03±2.6	53.24±2.9	47.68±2.5	35.92±2.8	72.89±3.1	66.31±2.7	69.00±3.0	69.99±2.7	66.45±2.8	69.32±2.3	**72.95±2.6**
Jaf	80.53±6.3	78.95±6.6	66.32±6.4	52.63±5.9	77.89±6.0	84.21±6.1	85.26±6.3	87.37±6.2	88.42±5.9	86.84±6.0	**89.47±6.1**
Average	46.77	41.29	35.47	23.47	37.75	50.30	51.78	52.45	51.61	52.33	**54.75**

TABLE VII

COMPARISON OF KAPPA COEFFICIENT OF DIFFERENT ALGORITHMS WITH SVM. THE BEST RESULT FOR EACH DATASET IS IN BOLD.

Dataset	OriFea	AE	StackAE	LBP	HOG	TT	LRA	DLR	LSF-PC	DPC	TLFL
Y	0.2438	0.2109	0.2145	0.1059	0.1027	0.2894	0.3017	0.2964	0.3161	0.3197	**0.3244**
A	0.4141	0.2675	0.1993	0.1768	0.3889	0.4969	0.5163	0.5179	0.5173	0.5284	**0.5496**
P05	0.5463	0.4767	0.3862	0.2090	0.3794	0.5532	0.5597	0.5646	0.5328	0.5576	**0.5903**
P07	0.4761	0.4416	0.3683	0.1976	0.2635	0.5117	0.5475	0.5562	0.5198	0.5439	**0.5789**
P09	0.4478	0.4284	0.3779	0.1970	0.3045	0.5113	0.5327	0.5296	0.5228	0.5186	**0.5539**
P27	0.5359	0.4434	0.3877	0.2365	0.3802	0.5294	0.5467	0.5374	0.5493	0.5472	**0.5769**
P29	0.4448	0.4106	0.3645	0.2000	0.3015	0.4473	0.4421	0.4800	0.4643	0.4511	**0.4912**
lfw	0.0658	0.0541	0.0301	0.0104	0.0701	0.1047	0.1174	0.1095	0.1240	0.1316	**0.1388**
Cas	0.6290	0.5294	0.4739	0.3569	0.7212	0.6599	0.6872	0.6922	0.6611	0.6896	**0.7241**
Jaf	0.7836	0.7661	0.6257	0.4737	0.7544	0.8246	0.8363	0.8596	0.8713	0.8538	**0.8830**

- $numE$ denotes the number of iterations, which is range from 1 to 200.

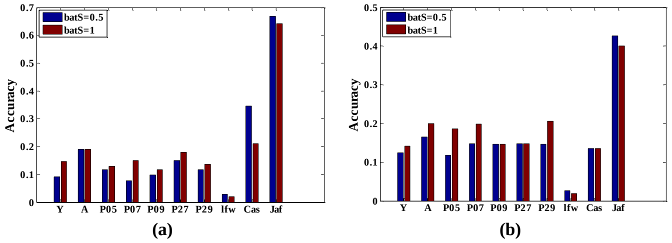

Fig. 11. Parameter batS in CNN. (a) Classifier is softmax. (b) Classifier is SVM.

Fig. 11 show the influence of parameter $batS$ on the performance of CNN. According to Fig. 11, we set the parameter $batS$ to 0.5 for dataset lfw, Cas and Jaffe, which we found to have the better performance. For other datasets, $batS$ is set to 1.

Fig. 12 and Fig. 13 give the plot of accuracy versus the declining rate $alpha$. The effect of parameter OM and KS on CNN can be observed from Fig. 14 and Fig. 15. Classifier softmax and SVM are used in Fig. 14 and Fig. 15, respectively. The number of iterations $numE$ is set such that the best accuracy of CNN is obtained. The five parameters used in two-layer CNN are adjusted to be the best for different datasets. Different parameter settings in CNN for various datasets are listed in Table VIII. The best results of two-layer CNN are represented in Table IX, from which we can see that the proposed two-layer framework TLFL can achieve superior performance than two-layer CNN.

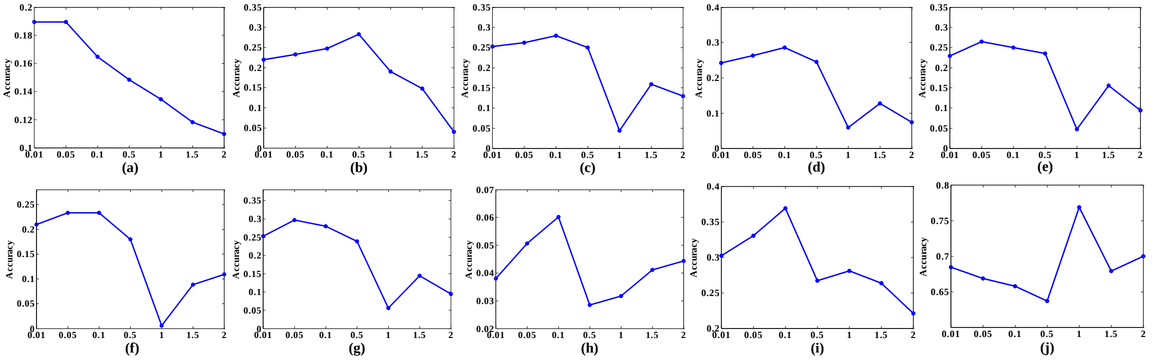

Fig. 12. Parameter alpha in CNN. Classifier is softmax. (a) Y. (b) A. (c) P05. (d) P07. (e) P09. (f) P27. (g) P29. (h) lfw. (i) Cas. (j) Jaf.

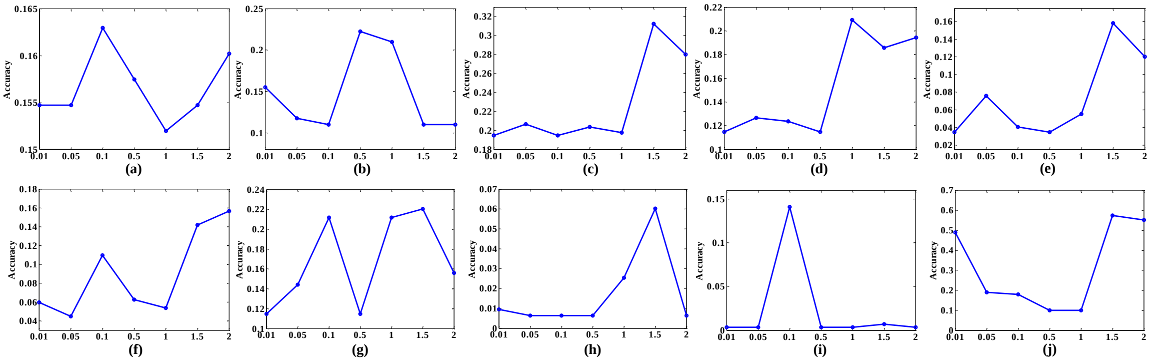

Fig. 13. Parameter alpha in CNN. Classifier is SVM. (a) Y. (b) A. (c) P05. (d) P07. (e) P09. (f) P27. (g) P29. (h) lfw. (i) Cas. (j) Jaf.

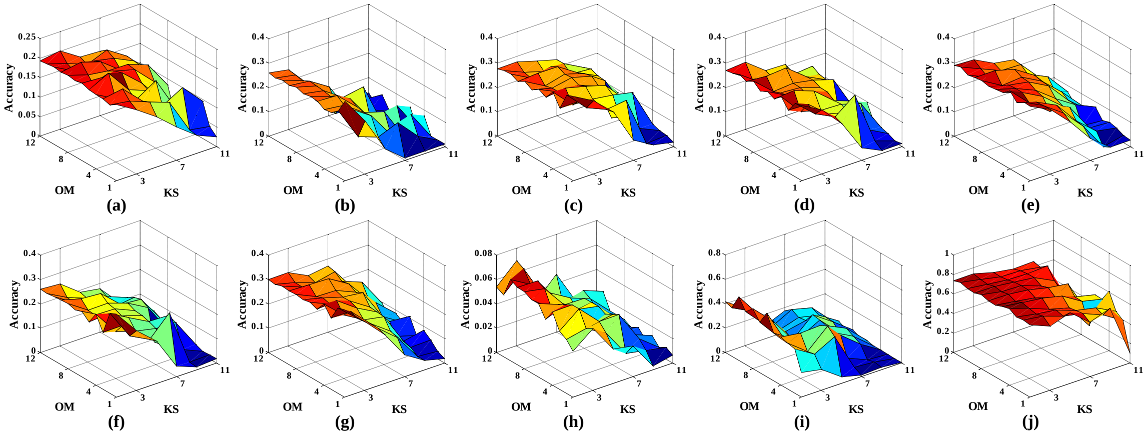

Fig. 14. Parameter OM and KS in CNN. Classifier is softmax. (a) Y. (b) A. (c) P05. (d) P07. (e) P09. (f) P27. (g) P29. (h) lfw. (i) Cas. (j) Jaf.

TABLE VIII
PARAMETER SETTINGS OF CNN FOR DIFFERENT DATASETS.

Dataset	softmax					SVM				
	batS	alpha	OM	KS	numE	batS	alpha	OM	KS	numE
Y	1	0.01	5	3	200	1	0.1	1	1	20
A	1	0.5	1	1	5	1	0.5	1	1	200
P05	1	0.1	1	1	50	1	1.5	1	1	100
P07	1	0.1	1	1	50	1	1	1	3	100
P09	1	0.05	4	1	50	1	1.5	6	3	100
P27	1	0.1	1	1	50	1	2	1	3	100
P29	1	0.05	1	1	50	1	1.5	1	1	100
lfw	0.5	0.1	12	3	150	0.5	1.5	7	7	100
Cas	0.5	0.1	6	1	150	0.5	0.1	3	1	150
Jaf	0.5	1	4	1	80	0.5	1.5	5	3	50

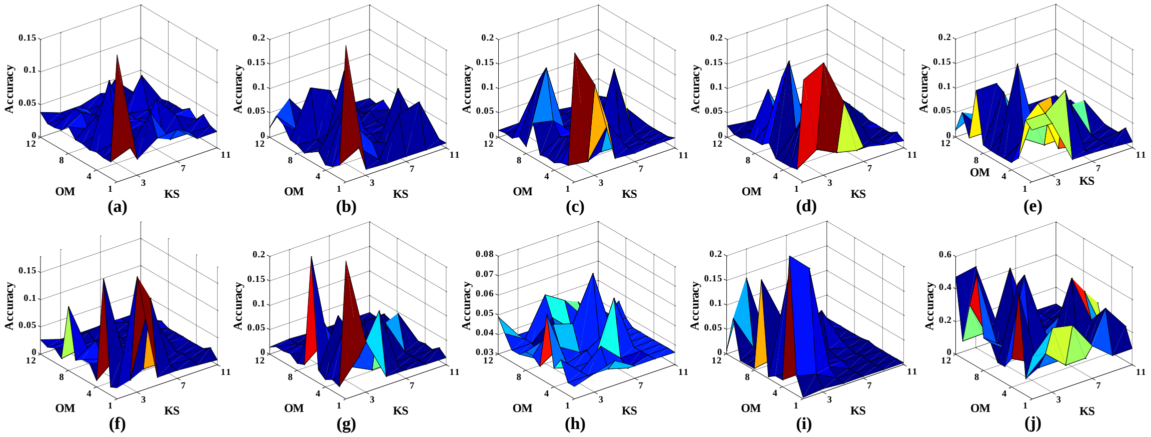

Fig. 15. Parameter OM and KS in CNN. Classifier is SVM. (a) Y. (b) A. (c) P05. (d) P07. (e) P09. (f) P27. (g) P29. (h) lfw. (i) Cas. (j) Jaf.

V. Conclusion

In this paper, a novel two-layer local-to-global feature learning framework TLFL is proposed for single sample per person face recognition. TLFL makes use of the advantages of both local features and global features. In the first layer of TLFL, local features are selected by a patch-based fuzzy rough set feature selection strategy. Global features are then extracted from the collected local features by a sparse autoencoder in the second layer. The quality of the final features are examined through softmax and SVM classifiers. The effectiveness of our proposed algorithm is demonstrated through a series of experiments on several face datasets.

The TLFL algorithm is a supervised method. Considering there are many unlabeled samples in practical applications, it is worth looking into whether unlabeled sample information can be exploited to further improve the performance of the TLFL algorithm.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments and constructive suggestions for this paper.

TABLE IX
Comparison of classification accuracies of two-layer CNN with TLFL. The best result with different classifiers is in bold. (%)

Dataset	softmax		SVM	
	CNN	TLFL	CNN	TLFL
Y	21.70	**31.27**	19.51	**34.86**
A	32.75	**51.00**	28.00	**55.50**
P05	34.71	**57.06**	26.47	**59.71**
P07	30.97	**45.13**	23.01	**54.87**
P09	32.06	**49.12**	18.53	**56.47**
P27	34.22	**54.57**	21.24	**58.70**
P29	35.88	**43.53**	28.24	**49.71**
lfw	6.96	**14.52**	4.11	**15.21**
Cas	52.46	**66.13**	27.46	**72.95**
Jaf	78.42	**88.26**	62.11	**89.47**

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *Acm Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[2] J. Daugman, "Face and gesture recognition: Overview," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 675–676, 1997.

[3] S. M. Azeem A, "A survey: Face recognition techniques under partial occlusion," *International Arab Journal of Information Technology*, vol. 11, no. 1, pp. 1–10, 2014.

[4] R. Min, A. Hadid, and J. L. Dugelay, "Improving the recognition of faces occluded by facial accessories," in *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, 2011, pp. 442–447.

[5] C. X. Ren, Z. Lei, D. Q. Dai, and S. Z. Li, "Enhanced local gradient order features and discriminant analysis for face recognition." *IEEE Transactions on Cybernetics*, 2015.

[6] Q. Feng, C. Yuan, J. S. Pan, J. F. Yang, Y. T. Chou, Y. Zhou, and W. Li, "Superimposed sparse parameter classifiers for face recognition." *Cybernetics IEEE Transactions on*, pp. 1–13, 2016.

[7] N. Mclaughlin, J. Ming, and D. Crookes, "Largest matching areas for illumination and occlusion robust face recognition." *Cybernetics IEEE Transactions on*, pp. 1–13, 2016.

[8] Forczmanski, *Recognition of Occluded Faces Based on Multi-subspace Classification*. Springer Berlin Heidelberg, 2013.

[9] ——, *Improving the Recognition of Occluded Faces by Means of Two-dimensional Orthogonal Projection into Local Subspaces*. Springer International Publishing, 2015.

[10] Jolliffe and Ian, "Principal component analysis," *Springer Berlin*, vol. 87, no. 100, pp. 41–64, 1986.

[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[12] T. Ahonen and A. Hadid, *Face Recognition with Local Binary Patterns*. Springer Berlin Heidelberg, 2004.

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

[14] X. Tan, S. Chen, Z. H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.

[15] H. K. Ji, Q. S. Sun, Z. X. Ji, Y. H. Yuan, and G. Q. Zhang, "Collaborative probabilistic labels for face recognition from single sample per person," *Pattern Recognition*, vol. 62, pp. 125–134, 2016.

[16] Y. Su, S. Shan, X. Chen, and W. Gao, "Adaptive generic learning for face recognition from a single sample per person," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2699–2706.

[17] W. Deng, J. Hu, and J. Guo, "Extended src: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–70, 2012.

[18] L. Zhuang, T. H. Chan, A. Y. Yang, S. S. Sastry, and Y. Ma, "Sparse illumination learning and transfer for single-sample face recognition with image corruption and misalignment," *International Journal of Computer Vision*, vol. 114, no. 2, pp. 272–287, 2015.

[19] D. Zhang, S. Chen, and Z. H. Zhou, "A new face recognition method based on svd perturbation for single example image per person," *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 895–907, 2005.

[20] Q. X. Gao, L. Zhang, and D. Zhang, "Face recognition using flda with single training image per person," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 726–734, 2008.

[21] H. Mohammadzade and D. Hatzinakos, "Projection into expression subspaces for face recognition from single sample per person," *IEEE Transactions on Affective Computing*, vol. 4, no. 1, pp. 69–82, 2013.

[22] C. Hu, M. Ye, S. Ji, W. Zeng, and X. Lu, "A new face recognition method based on image decomposition for single sample per person problem," *Neurocomputing*, vol. 160, no. C, pp. 287–299, 2015.

[23] J. Lu, Y. P. Tan, and G. Wang, "Discriminative multimanifold analysis for face recognition from a single training sample per person," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 1943–1950, 2013.

[24] A. M. Mart and nez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748–763, 2002.

[25] Z. L. Sun and L. Shang, "A local spectral feature based face recognition approach for the one-sample-per-person problem," *Neurocomputing*, vol. 188, pp. 160–166, 2016.

[26] T. Pei, L. Zhang, B. Wang, F. Li, and Z. Zhang, "Decision pyramid classifier for face recognition under complex variations using single sample per person," *Pattern Recognition*, 2016.

[27] F. Liu, J. Tang, Y. Song, Y. Bi, and S. Yang, "Local structure based multi-phase collaborative representation for face recognition with single sample per person," *Information Sciences*, vol. s 346šC347, pp. 198–215, 2016.

[28] J. Lu, Y. P. Tan, and G. Wang, "Discriminative multimanifold analysis for face recognition from a single training sample per person," in *International Conference on Computer Vision*, 2011, pp. 1943–1950.

[29] S. Gao, K. Jia, L. Zhuang, and Y. Ma, "Neither global nor local: Regularized patch-based representation for single sample per person face recognition," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 365–383, 2015.

[30] W. K. Wong and M. Sun, "Deep learning regularized fisher mappings," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, p. 1668, 2011.

[31] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 8, pp. 1887–1901, 2013.

[32] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *Neural Networks and Learning Systems IEEE Transactions on*, vol. 25, no. 12, pp. 2303–2308, 2014.

[33] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[34] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," pp. 384–392, 2015.

[35] G. Goswami, R. Bhardwaj, R. Singh, and M. Vatsa, "Mdlface: Memorability augmented deep learning for video face recognition," in *IEEE International Joint Conference on Biometrics*, 2014, pp. 1–7.

[36] L. Tian, C. Fan, Y. Ming, and Y. Jin, "Stacked pca network (spcanet): An effective deep learning for face recognition," in *IEEE International Conference on Digital Signal Processing*, 2015, pp. 1039–1043.

[37] S. Gao, Y. Zhang, K. Jia, and J. Lu, "Single sample face recognition via learning deep supervised autoencoders," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 1–1, 2015.

[38] P. J. S. Vega, R. Q. Feitosa, V. H. A. Quirita, and P. N. Happ, "Single sample face recognition from video via stacked supervised autoencoder," in *Graphics, Patterns and Images*, 2016, pp. 96–103.

[39] J. Dai and Q. Xu, "Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification," *Applied Soft Computing*, vol. 13, no. 1, pp. 211–221, 2013.

[40] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, p. 119, 2016.

[41] H. C. Shin, M. R. Orton, D. J. Collins, and S. J. Doran, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data," *IEEE Transactions on Software Engineering*, vol. 35, no. 8, pp. 1930–1943, 2013.

[42] Y. Xu, L. Yao, D. Zhang, and J. Y. Yang, "Improving the interest operator for face recognition," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9719–9728, 2009.

[43] H. L. Chen, B. Yang, J. Liu, and D. Y. Liu, "A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis," *Expert Systems with Applications An International Journal*, vol. 38, no. 7, pp. 9014–9022, 2011.

[44] K. Thangavel and A. Pethalakshmi, "Dimensionality reduction based on rough set theory: A review," *Applied Soft Computing*, vol. 9, no. 1, pp. 1–12, 2009.

[45] J. Jelonek, K. Krawiec, and R. Slowišœski, "Rough set reduction of attributes and their domains for neural networks," *Computational Intelligence*, vol. 11, no. 2, pp. 339–347, 1995.

[46] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[47] A. M. Martinez, "The ar face database," *Cvc Technical Report*, vol. 24, 1998.

[48] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *Pattern Analysis and Machine Intelligence IEEE Transactions on*, vol. 25, no. 12, pp. 1615 – 1618, 2003.

[49] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," 2008.

[50] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations," *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, vol. 38, no. 1, pp. 149–161, 2008.

[51] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings*, 1998, pp. 200–205.

[52] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 1–7, 2007.

[53] W. Deng, J. Hu, X. Zhou, and J. Guo, "Equidistant prototypes embedding for single sample based face recognition with generic learning and incremental learning," *Pattern Recognition*, vol. 47, no. 12, pp. 3738–3749, 2014.

[54] F. Yin, L. C. Jiao, F. Shang, L. Xiong, and S. Mao, "Double linear regressions for single labeled image per person face recognition," *Pattern Recognition*, vol. 47, no. 4, pp. 1547–1558, 2014.

[55] H. Yan, J. Lu, X. Zhou, and Y. Shang, "Multi-feature multi-manifold learning for single-sample face recognition," *Neurocomputing*, vol. 143, no. 16, pp. 134–143, 2014.

[56] K. Chen, J. Hu, and J. He, "A framework for automatically extracting overvoltage features based on sparse autoencoder," *IEEE Transactions on Smart Grid*, pp. 1–1, 2016.

[57] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "One-shot learning for real-time action recognition," *Lecture Notes in Computer Science*, vol. 7887, pp. 31–40, 2013.

[58] Q. Hu, D. Yu, W. Pedrycz, and D. Chen, "Kernelized fuzzy rough sets and their applications," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 11, pp. 1649–1667, 2011.

[59] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[60] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions." *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 168–182, 2010.

[61] R. B. Palm, "Deeplearntoolbox," https://github.com/rasmusbergpalm/DeepLearnToolbox.

Yuwei Guo was born in China in 1988. She received the B.S. degree from Youdian University, Xi'an, China, in 2010. Since then, she has been taking successive postgraduate and doctoral programs with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, International Research Center of Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, Xidian University, Xi'an, China.

Her research interests include rough set theory, data mining and deep learning.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China, International Research Center of Intelligent Perception and Computation.

His current research interests include intelligent information processing, image processing,machine learning, and pattern recognition. Prof. Jiao is a member of the IEEE Xi'an Section Execution Committee; the President of the Computational Intelligence Chapter, the IEEE Xi'an Section, and the IET Xi'an Network; the Chairman of the Awards and Recognition Committee; the Vice Board Chairperson of the Chinese Association of Artificial Intelligence; a Councilor of the Chinese Institute of Electronics; a Committee Member of the Chinese Committee of Neural Networks; and an Expert of the Academic Degrees Committee of the State Council. Prof. Jiao was a recipient of the Second Prize of the National Natural Science Award in 2013.

Shuang Wang (M'07), was born in Shannxi, China, in 1978. She received the B.S., M.S., and Ph.D. degrees in circuits and systems form Xidian University, Xi'an, China, in 2000 and 2003, respectively. Currently, she is a Professor in the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University.

Her main research interests are Sparse Representation, image processing and high-resolution SAR image processing.

Shuo Wang is a Research Fellow at the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) in the School of Computer Science, the University of Birmingham (UK). She received the B.Sc. degree in Computer Science from the Beijing University of Technology (BJUT), China, in 2006, and was a member of Embedded Software and System Institute in BJUT in 2007. She received the Ph.D. degree in Computer Science from the University of Birmingham, U.K., in 2011, sponsored by the Overseas Research Students Award (ORSAS) from the British Government (2007). Dr. Wang has worked on 3 EPSRC-funded projects and 2 EU-funded projects. She is currently a project theme leader of EPSRC-funded project.

Her research interests include class imbalance learning, ensemble learning, online learning and machine learning in software engineering. Her work has been published in internationally renowned journals and conferences.

Fang Liu (SM'07) received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1984, and the M.S. degree from Xidian University, Xi'an, in 1995, both in computer science and technology. She is currently a Professor with Xidian University. She has authored or co-authored five books and over 80 papers in journals and conferences.

Her current research interests include image perception and pattern recognition, machine learning, and data mining. Prof. Liu was a recipient of the Second Prize of the National Natural Science Award in 2013.