Fuzzy Nearest Feature Line-based Manifold Embedding for Facial Expression Recognition*

WEI LI, QIUQI RUAN AND JUN WAN
Institute of Information Science
Beijing Jiaotong University
Beijing, 100044 P.R. China

Traditional nearest feature line (NFL) based subspace learning (NFLS) has been successfully applied in face recognition by capturing more variations of face images than original data set. However, it is suboptimal for pattern recognition due to that it only focuses on the within-class information while neglects the interaction between different classes. Besides, NFLS fails to detect the underlying manifold embedded in image space. In this paper, a novel manifold embedded extension of NFLS named fuzzy local nearest feature line (FLNFL) is developed for facial expression recognition. FLNFL achieves its high classification performance by combining graph embedded projection with fuzzy set theory in a Fisher type of learning manner. It not only considers the intrinsic similarity between data points, but also exploits fuzzy assignment technique to investigate inherent uncertainties in expression images arise from variety of illumination, posture, and viewing directions. After this mapping, therefore, the transformed features can reflect both local geometry and intrinsic class assignment of the data set. Experimental results on two widely used facial expression databases suggest that the proposed method provides a better representation of expression features and yields promising performance in facial expression recognition.

Keywords: nearest feature line, fuzzy K-nearest neighbor, manifold embedding, discriminant analysis, facial expression recognition

1. INTRODUCTION

Facial expression, which is one of the most natural and intuitive means for human beings to communicate their emotions and intentions, succeeding in conveying plenty of information people can not express by languages, has been widely studied to realize human-computer intelligent interaction. Potential benefits from facial expression recognition include distance education learning, automobile driver alertness monitoring, lie detection, affect assessment in health, and for constructing more entertaining and interactive games [1]. All these applications make facial expression recognition an exciting research area [2, 3].

Facial expression recognition is very difficult and challenging due to the fact that expression varies a lot between different individuals. Moreover, it may be affected by some latent variables, i.e. illumination, pose and facial deformations. In 1971 American psychologists Ekman and Friesen defined six kinds of basic expressions: happiness, sadness, fear, anger, disgust and surprise [4]. Existing expression analyzers are usually designed based on this classification. Generally speaking, facial expression recognition
includes three phases, *i.e.* facial region detection and pre-processing, expression feature extraction and classification. The most fundamental issue of facial expression recognition is that how to extract effective expression features from original face images for classification. In real pattern recognition tasks, data samples generally have high dimension and small sample size. Direct operations on these data may result in the curse of dimensionality [5] as well as computational complexity. Consequently, dimensionality reduction is an essential step in many engineering applications, usually utilized for data simplification.

Two most well-known conventional dimensionality reduction methods are Principal Component Analysis (PCA) [6] and Linear Discriminant Analysis (LDA) [7, 8]. PCA provides an optimal linear transformation from the original image space to an orthogonal eigenspace with reduced dimensionality in the sense of least mean squared reconstruction error. In contrast with the unsupervised method of PCA, LDA is a supervised technique which seeks for the projection axes on which the data points have maximum between-class separation and minimum within-class dispersion and is widely believed to be able to enhance class separability. However, the classification performance of traditional LDA is suboptimal due to the fact that it doesn’t consider the effect of different distances between different classes [9]. Close classes in original space will merge in the projected subspace, while the already far apart classes could be preserved as much as possible to meet the objective for subspace selection, and thus can potentially result in misclassification. A number of regularization techniques that might alleviate this problem have been proposed. For instance, Lotlikar *et al.* [10] introduced fractional-step into linear discriminant analysis, where the dimensionality reduction is implemented in a series of fractional steps allowing for the relevant distances to be more properly weighted. aPAC [11] boosts its discriminative power by employing a weighting scheme derived from approximating the Bayes error for pairs of classes. Recently, it has been shown that this problem can be mitigated to some extent by using the general mean criteria, such as geometric mean (MGMD) [9, 12] and harmonic mean (MHMD) [13], to replace the arithmetic mean used in LDA. These general mean criteria are adaptive, since they amplify the effects of the small distance class pairs and, simultaneously reduce the effects of the well separated class pairs. Different from the aforementioned weighting schemes, MMDA directly maximizes the minimum distance of all class pairs in the low-dimensional subspace, and thus it duly considers the separation of all classes [14].

Another drawback of LDA is that it can not be implemented directly when the within-class scatter is singular due to the small sample size (SSS) problem that occurs frequently in practice whenever the dimensionality of data exceeds the sample size. A traditional solution is to incorporate PCA step into the LDA framework [15]. In this approach, PCA is used as a preprocessing step for dimensionality reduction so as to discard the null space of the within-class scatter matrix of the training data set. Then LDA is performed in the lower dimensional PCA subspace. In order to keep the null space of within-class scatter matrix, which carries discriminative information, Zhuang *et al.* [16] developed the inverse Fisher criterion under the modified PCA procedure. Yu *et al.* [17] reverse the whitening order of the within and between class scatter matrices, so that LDA can proceed without PCA step. In [18], the technique of regularization was used. It aims to smooth out the effects of ill or poorly conditioned covariance estimates. 2-D based scheme [19, 20] is also a good choice to avoid the SSS problem and, keep the important spatial structure information.
Recent studies show that images of a subject’s facial expressions define a smooth manifold hidden in the high dimensional image space [21, 22]. However, all extensions discussed above give good representations of the global Euclidean structure but fail to capture the structure lies on a nonlinear submanifold. Therefore, it is reasonable to apply manifold learning to reveal the intrinsic distribution of expression images. Representative manifold learning based algorithms include Locally Linear Embedding (LLE) [23], ISOAMP [24], Laplacian Eigenmap (LE) [25]. These methods have been shown capable in discovering geometrical structure of the underlying manifold, yet the negligence of label information implies that the discriminating power of these nonlinear embeddings cannot be guaranteed sufficiently high. Moreover, it cannot project a new testing point directly because of the implicitness of the nonlinear projection. This fact is the major cause of the practical limitations of these manifold embedding algorithms. To deal with this problem, a direct attempt is to employ linearization procedure to compute explicit maps over new measurements. Such operation is applied by Locality Preserving Projection (LPP) [26], a linearization of LE; Neighborhood Preserving Projection (NPP) [27], a linearization of LLE, and their variants (see [28-30]).

To the best of our knowledge, most existing dimensionality reduction methods suffer from the overfitting phenomenon [31]. In such problem, only a small number of training samples per class are available, inducing the subspaces learned by the feature extractors not to be used to extract good features for the purpose of classification. Nearest feature line (NFL) is an effective solvent to resist overfitting. It was proposed originally for classification by Li et al. [32, 33], and recently have been applied to represent faces in facial images analysis [31, 34]. The most favorable property of NFL is that it can extend the capacity of limited feature points by calculating a linear function to interpolate and extrapolate each feature pair belonging to the same class. Conventional classifiers treat the feature space as a set of isolated points and measure the distance between the query and an individual prototype, whereas the classification of NFL is based on the minimum distance between the query to the feature line (FL) connecting any two prototype feature points of the same class. If the sample number of a class is $N$, then $N(N – 1)/2$ FLs have to be constructed to represent this class, which is often much larger than $N$. The representational capacity of available training samples is thus enlarged. Pang et al. [31] first introduced NFL metric to feature extraction case and proposed NFLS. NFLS inherits the desirable virtue from NFL naturally, that is robust to overfitting and has been proved to be effective, yielding encouraging results on face recognition. However, NFLS is suboptimal for pattern recognition because it is modeled only to make the points of the same class cluster closer, a modeling which has no direct relation to classification. UDNFLA [34] is an extension of NFLS which incorporates the idea of discriminant analysis by considering both the intraclass and interclass information to design the discriminator. Intrinsically, it is a classification-oriented subspace learning and has been shown more powerful than NFLS. Nonetheless, both NFLS and UDNFLA are global structure based algorithms. They are incapable to explore the underlying local structure.

More recently, the concept of “fuzzification” has attracted increasing attention for class assignment in [35-37], following the early work [38]. The fuzzy K-nearest neighbor (FKNN) algorithm [38] assigns class membership to a measurement rather than assigning the measurement to a particular class. That is, it provides a means of defining categories that are inherently imprecise. This will benefit for the “blended” expressions and those
affected by the variety of environment illumination and face poses.

In this paper, to improve the classification performance of NFLS, we propose a novel manifold learning algorithm termed as fuzzy local NFL (FLNFL), which encodes with fuzzy set theory and graph embedded projection, and is effective to cope with the factor of uncertainty being inherently present in expression images arise from various variations. In our method, adjacent graph, weighting function, and statistical property dwell on the samples distribution information represented by fuzzy membership degrees [38]. After being embedding into a lower-dimensional subspace, data points from the same class cluster closer, whereas neighboring points from different class keep away from one another. It is worth stressing three major advantages of our proposed FLNFL:

1. Our method inherits the favorable virtue of NFL that can extend the capacity of limited feature points, and is robust to the case that when the training set is small.
2. While both of the NFLS and UDNFLA approaches aim to preserve the global structure of the data set, the proposed method considers the locality of measurements. Thus, it can deal with the underlying structure of expression manifold.
3. By resorting to the fuzzy set theory, the proposed method can effectively address the uncertainties arise from variations that conventional methods based on the concept of crisp (yes-no) class assignment meaning may fail to find the optimal direction for classification purpose.

The rest of the paper is organized as following: Those methods related to our approach, such as NFL, NFLS, and FKNN will be briefly introduced in section 2. Section 3 describes our proposed FLNFL algorithm. Section 4 verifies the effectiveness of our method through a variety experiments on two benchmark facial expression databases. Finally, conclusions are drawn in section 5.

2. BACKGROUND

Given a data set \( X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d \times N} \), where each column corresponds to an image sample belonging to one of the \( L \) classes \( C_1, C_2, \ldots, C_L \). Each class contains \( N_c \) \( (c = 1, 2, \ldots, L) \) samples. Denote the class label of \( x_i \) as \( l_i \). The generic problem of linear dimensionality reduction is to seek a transformation matrix \( V \in \mathbb{R}^{d \times d} \) that projects these \( N \) points to a set of points \( Y = [y_1, y_2, \ldots, y_N] \) in \( d \) dimensional space \( d \ll D \). That is,

\[
Y = V^T X.
\]

The transformation matrix \( V \) is achieved according to a different way refers to a particular type of objective function, varying from the desirable properties to be preserved. The properties include global structure and neighborhood geometry. For our proposed approach, the neighborhood geometry is the property what we focus on to deal with nonlinear manifold structure. It is advantageous to walk through the pertinent techniques that are used in our approach before describing our proposed method. Below, we discuss in detail the context of NFL, NFLS and FKNN algorithms.
2.1 NFL

NFL [32, 33] is a nonparametric classifier which aims to expand the representational capacity of available prototypes accounting for as many variations as possible. By taking use of the prior knowledge of prototypes, a subspace called FL space is constructed for each class, consisting of straight lines (called FLs) passing through each pair of feature points belonging to the same class. FL approximates variants of the two prototypes it connects, and virtually provides an infinite number of feature points of the class. NFL can thus capture more variations than the original data points. The basic assumption of NFL is that at least two distinct prototype feature points are available for each class, which is usually satisfied. Notice that the NFL algorithm is originally used as a classifier in feature space, the following discussion in this subsection is based on representations of original samples in transformed space.

Let $y_1$ and $y_2$ be any two distinct prototype features of the same class, a straight line passing through these points is called a FL of this class, denoted by $y_1 \hat{y}_2$. A FL covers more of the feature space than the two feature points alone. Take class $C$ for instance, a number of $FL_c = \frac{N_c(N_c - 1)}{2}$ FLs are constructed to represent the class, where $N_c (c = 1, 2, \ldots, L)$ denotes the sample number of class $C$. The generalization ability of this class is thus increased.

Suppose a testing point $y_m$, its projection point $y_n$ onto the feature line $\hat{y}_1\hat{y}_2$ is $y_n = y_1 + \nu(y_2 - y_1)$, $\nu$ is called the position parameter, which can be calculated by $\nu = \frac{(y_m - y_1)(y_2 - y_1)}{(y_2 - y_1)^T(y_2 - y_1)}$. NFL doesn’t compute the distance between $y_m$ and $y_1$, nor the distance between $y_m$ and $y_2$. Instead, the distance between $y_m$ and the projection point $y_n$ on the line $\hat{y}_1\hat{y}_2$, which is denoted by $d(y_m, \hat{y}_1\hat{y}_2)$, is used as the metric, as illustrated in Fig. 1. Where $d(y_m, \hat{y}_1\hat{y}_2) = ||y_m - y_n||$ is termed as the FL distance. For NFL, classification is based on the minimal FL distance. In contrast, class assignment of the nearest neighbor (NN) classifier depends on the nearest point among all prototype feature points. Obviously, the distance of query to FL is smaller than those between two prototype feature points (as shown in Fig. 1).

$$d(y_m, \hat{y}_1\hat{y}_2) \leq \min(d(y_m, y_1), d(y_m, y_2)) \tag{2}$$

![Fig. 1. Generalizing two prototype feature points $y_1$ and $y_2$ by the feature line $\hat{y}_1\hat{y}_2$. The feature point $y_m$ of the query point is projected onto the line as point $y_n$.](image-url)
That is, the class determined by NFL attains smaller distance than that determined by NN.

2.2 Nearest Feature Line for Subspace Learning (NFLS)

The property that extends the representational capacity of prototype features is also desirable for the feature extraction phase to resist the influence of overfitting. Motivated by this point, Pang et al. [31] generalized NFL for feature extraction (NFLS), which aims to find an optimal transformation \( V \in \mathbb{R}^{d \times d} \) such that the FL distances of the within-class are minimized after projection. The objective function to be minimized is:

\[
J(V) = \sum_{i=1}^{N} \sum_{j \in M(i)} \| V^T x_i - V^T x_j \|^2
\]

\[
= \sum_{i=1}^{N} \sum_{j \in M(i)} \text{tr}[V^T (x_i - x_j)(x_i - x_j)^T V]
\]

\[
= \text{tr}[V^T \sum_{i=1}^{N} \sum_{j \in M(i)} ((x_i - x_j)(x_i - x_j)^T) V]
\]

\[
= \text{tr}[V^T S_L V],
\]

where \( x_j \) denotes the projection point of \( x_i \) onto FLs, \( M(i) = \{ j \mid k(x_i) = k(x_j) \} \) represent the projection points with the same class label as \( x_i \), \( S_L \) is defined as:

\[
S_L = \sum_{i=1}^{N} \sum_{j \in M(i)} (x_i - x_j)(x_i - x_j)^T.
\]

In order to remove the arbitrary scaling factor in the embedding, NFLS imposes a constraint as follows:

\[
V^TV = I_d,
\]

where \( I_d \) is a \( d \times d \) identity matrix.

Finally, the transformation matrix \( V \) that minimizes the objective function is obtained by solving the eigenvalue problem:

\[
S_L V_l = \lambda_l V_l.
\]

2.3 FKNN

FKNN [38] provides a means of defining categories that are inherently imprecise by classifying a sample vector quantitatively into multiple categories rather than assigning the vector to a particular class. In a broad sense, quantifying the degree of class membership is more appropriate to explore the intrinsic class assignment. The reasons are twofold: First, specifying the degree of membership in a set, rather than just the binary is or isn’t a member enables us to differentiate overlapping samples, such as blended expressions. Second, since expression images in general suffer numerous environmental factors, such as illumination conditions, poses or perspective etc., the information of
class membership degree is beneficial to investigate these factors and express the uncertainty in data set. Therefore, the soft assignment technique is expected to improve performance in classification task.

Recall the data matrix $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{D \times N}$, a fuzzy $K$ partition of these vectors specifies the degree of membership of each vector in each of $L$ classes. It is denoted by the $L \times N$ matrix $U$, where $u_{ij} = u(x_j)$ for $i = 1, ..., L$ and $j = 1, ..., N$ is the degree of membership of $x_j$ in class $i$. FKNN imposes two constraints on $U$ for mathematical tractability:

$$\sum_{i=1}^{L} u_{ij} = 1, \quad (7)$$

$$0 < \sum_{j=1}^{N} u_{ij} < N. \quad (8)$$

According to [38], the membership degree matrix $U$ can be computed through four steps:

**Step 1:** Compute the Euclidean distance matrix between pairs of feature vectors in the training set.

**Step 2:** Set diagonal elements of this matrix to infinity.

**Step 3:** Sort the distance matrix (treat each of its columns separately) in an ascending order. Collect the corresponding class labels of the patterns located in the closest neighborhood of the pattern under consideration (as we are concerned with “$K$” neighbors, this return a list of “$K$” integers).

**Step 4:** Compute the membership grade to class “$i$”, for $j$th pattern using the expression:

$$u_{ij} = \begin{cases} 0.51 + 0.49/(n_{ij}/K) & \text{if } i = \text{the same as the label of the } j\text{th pattern} \\ 0.49/(n_{ij}/K) & \text{if } i \neq \text{the same as the label of the } j\text{th pattern} \end{cases} \quad (9)$$

where $n_{ij}$ denotes the number of the neighbors of the $j$th pattern that belong to the $i$th class. $K$ is the number of neighbors.

### 3. FUZZY LOCAL NEAREST FEATURE LINE

As discussed in section 1, images of a subject’s facial expressions define a smooth manifold hidden in the high dimensional image space. We generalize NFLS to graph embedding to elevate its classification performance. Rather than using conventional *crisp* class assignment, we exploit FKNN algorithm to construct adjacent graphs and compute statistical properties based on this redefined data distribution. By this way, the proposed method can effectively address overlapping samples and the uncertainties arise from variations. The basic objective of the proposed method is to seek a mapping of a given
data set so as to preserve the within-class local topology while reducing the overlap between neighboring classes. This bears an analogy to the Fisher criterion [7], which maximizes between-class separation and minimizes within-class dispersion.

3.1 The Objective Function of LNFL

The following section is based on fuzzy set theory. See section 2.3 for details and [38] for a comprehensive reference.

In the embedding design, FL distance is utilized to characterize the intra locality. First, we construct an adjacent graph $\mathcal{G}$ for a given sample $x_i$ by connecting within-class neighbor pairs from fuzzy $K$ nearest neighbors of $x_i$. $\mathcal{G}$ is termed as FL graph, and edges in $\mathcal{G}$ are the so called FLs, denoted by $\{e_{ij}\}_{i,j=1}^k$, where $l_i$ is the number of FLs corresponding to $x_i$. Note that $l_i$ varies with different $x_i$. If there are $k_i$ within-class neighbors for $x_i$, then $l_i = k_i(k_i - 1)/2$. By $P(x_i) = \{x_{ij}\}_{j=1}^{k_i}$ we denote the projection points of $x_i$ onto FLs.

A reasonable criterion to keep neighboring points with the same label close is to minimize the following function:

$$J_1(V) = \sum_{i=1}^{N} \sum_{j \in P(x_i)} w_{ij}^d ||V^T x_i - V^T x_j||^2$$

$$= \sum_{i=1}^{N} \sum_{j \in P(x_i)} \text{tr}[V^T w_{ij}^d \cdot (x_i - x_j)(x_i - x_j)^TV]$$

$$= \text{tr}[V^T \sum_{i=1}^{N} \sum_{j \in P(x_i)} w_{ij}^d \cdot (x_i - x_j)(x_i - x_j)^TV]$$

$$= V^T A V$$

where we refer to the matrix

$$A = \sum_{i=1}^{N} \sum_{j \in P(x_i)} w_{ij}^d (x_i - x_j)(x_i - x_j)^T$$

as the within-class scatter matrix. $w_{ij}^d$ is a weighting factor that depends on the fuzzy membership degrees of within-class neighbors of $x_i$:

$$w_{ij}^d = \frac{\mu_y(x_i)}{\sum_{j=1}^{k_i} \mu_y(x_j)}$$

The use of weighting factor $w_{ij}^d$ is based on the following geometric intuitions. It has to put more weights on points with high prior probabilities so as to guarantee that if $x_i^C$ and $x_j^C$ are close, then $y_i^C$ and $y_j^C$ are close as well.

Meanwhile, we expect to maximize the margin between neighboring classes. Considering the fuzzy membership degrees, the mean vector of each class $\tilde{m}_j$ is redefined as:
\[ \tilde{m}_j = \frac{\sum_{i=1}^{N} u_{ij}x_i}{\sum_{i=1}^{N} u_{ij}}. \]  

(13)

In the definition of between-class scatter matrix, a key point to note is to prevent outlier classes from dominating the sample distribution. Motivated by [10], a weighting function is introduced; its components can be defined as:

\[ w_{ij} = \exp(-\frac{||\tilde{m}_i - \tilde{m}_j||^2}{t}) \]

(14)

where \( t \) is a parameter.

The generalized between-class scatter matrix to be maximized can be expressed as:

\[ J_2(V) = \sum_{i,j=1}^{L} (m_i - m_j)^T w_{ij} \]

(15)

where \( m_i \) and \( m_j \) are separately the mean vectors for the \( i \)th class and \( j \)th class, i.e.

\[ m_i = \frac{1}{N_i} \sum_{k=1}^{N_i} y_{ik}. \]

(16)

By simple algebra formulation, \( J_2(V) \) can be reduced to

\[
\begin{align*}
\frac{1}{2} \sum_{i,j=1}^{L} (m_i - m_j)^T w_{ij} &= \frac{1}{2} \sum_{i,j=1}^{L} \left( \frac{1}{N_i} \sum_{k=1}^{N_i} x_{ik} \right)^2 w_{ij} + \frac{1}{2} \sum_{i,j=1}^{L} \left( \frac{1}{N_j} \sum_{k=1}^{N_j} y_{jk} \right)^2 w_{ij} \\
&= \frac{1}{2} \sum_{i,j=1}^{L} \left[ \left( \frac{1}{N_i} \sum_{k=1}^{N_i} x_{ik} \right)^2 - \left( \frac{1}{N_j} \sum_{k=1}^{N_j} y_{jk} \right)^2 \right] w_{ij} \\
&= \frac{1}{2} \sum_{i,j=1}^{L} \left[ \left( \frac{1}{N_i} \sum_{k=1}^{N_i} x_{ik} \right)^2 \right] w_{ij} - \left( \frac{1}{N_j} \sum_{k=1}^{N_j} y_{jk} \right)^2 w_{ij} \\
&= V^T F (E - W_{ij}) F^T V \\
&= V^T F H F^T V \\
&= V^T B V
\end{align*}
\]

where \( F = [\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_L] \); \( E \) is a diagonal matrix, and its entries are column (or row, since \( w_{ij} \) is symmetric) sum of \( w_{ij} \), \( E_{ii} = \sum_j w_{ij} \); \( H = E - W_{ij} \). \( B \) is the between-class scatter denoted as:

\[ B = F H F^T. \]

(18)
Now, we have the two optimization problems presented as Eqs. (10) and (15). Considering them simultaneously, we can obtain just such a projection by maximizing the following criterion:

$$\arg\max_{\gamma} \left( J_{\gamma}(V) - J_{\gamma}(V) \right).$$

To uniquely determine $V$, the constraint $V^T V = I_d$ is imposed on Eq. (19). That is identical to that reported in [31]. Then the objective function of FLNFL can be formulated as:

$$\arg\max \ { \text{tr} } \{ V^T (B - A)V \}$$

$$\text{s.t.} \ V^T V = I_d,$$

(20)

Obviously, the solutions of Eq. (20) are obtained by solving a standard eigen-decomposition problem:

$$(B - A)v_i = \lambda_i V_i.$$

(21)

The optimal solution for Eq. (21) is the eigenvectors associated with $d$ largest eigenvalues. Once $V$ is obtained, the linear features are extracted by $V^T X$.

3.2 The Algorithm of FLNFL

In real applications of such facial expression recognition, the standard eigen Eq. (21) sometimes has no solutions since the difference matrix $(B - A)$ is singular due to the limited number of training samples. To tackle with this computing problem, a PCA preprocessing step is exploited to remove its null space. In this section, we summarize the steps of the FLNFL algorithm.

**Step 1:** Calculate $S_i$’s $m$ largest eigenvalues and the associated $m$ orthogonal eigenvectors $\phi_1, \phi_2, \ldots, \phi_m$ using the approach presented in [6]. Let $V_{PCA} = [\phi_1, \phi_2, \ldots, \phi_m]$, then we get $Y = V_{PCA} X$, where $X = [x_1, x_2, \ldots, x_N]$.

**Step 2:** Assign $K$ neighbors for each sample, then compute fuzzy membership degree matrix $U$ and class center matrix $F$ through FKNN algorithm.

**Step 3:** Select all the within-class neighbors of each sample from its fuzzy $K$-nearest neighbors to form FL graph $\zeta$. Projection points onto FLs can be achieved with NFL algorithm.

**Step 4:** According to $U$ and $F$ work out within-class scatter matrix $A$ and between-class scatter matrix $B$ from Eqs. (11) and (18) respectively.

**Step 5:** Solve the standard eigenvalue problem $(B - A)v_i = \lambda_i V_i$ to obtain the LNFL pro-
projection axes $V_{FLNFL} = [v_1, v_2, \ldots, v_d]$, which corresponds to the $d$ largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$.

**Step 6:** Project all samples onto the obtained optimal discriminant vectors through $x_i \rightarrow z_i = V^T x_i$, $V = V_{PCA} \cdot V_{FLNFL}$.

**4. EXPERIMENTS AND DISCUSSION**

**4.1 Dataset**

In this section, we assess the feasibility and performance of our proposed FLNFL on facial expression recognition task, using two public databases, the JAFFE database and the Cohn-Kanade database.

The JAFFE database contains 219 images of 10 Japanese females with 3 or 4 examples of each of the six primary emotion expressions and a neutral face. A random subset with 21 per expression was chosen as our database and totally 147($7 \times 21$) samples were used in our experiments. We divided the subset into two cases to form training and testing set:

Case 1: number of training set for each class: 20, test set: 1;
Case 2: number of training set for each class: 19, test set: 2.

For each case, we averaged the results over 10 random splits. Fig. 2 (a) shows some samples in JAFFE, where each sample is resized to $64 \times 64$ based on the eye locations.

Image data of Cohn-Kanade database consist of approximately 500 image sequences from 100 subjects range from 18 to 30 years. 65 percent were female; 15 percent were African-American and 3 percent Asian or Latio. We selected samples of the six primary expressions and a neutral face from 25 subjects of the database. 4 samples of different intensity of each subject, and totally 700($7 \times 4 \times 25$) samples were used in our experiments. All images were cropped to a size of $25 \times 25$ based on eye locations, resulting in a 625-dimensional input space. Fig. 2 (b) depicts a set of images of one person from the Cohn-Kanade database. The training and testing set are selected randomly by choosing for each subject two cases:

Case 1: number of training set for one person: 2, test set: 2;
Case 2: number of training set for one person: 3, test set: 1.

![Image](image.png)

**Fig. 2.** Samples of facial expression image in two databases.
To evaluate the proposed algorithm, we systematically compare the proposed FLNFL with PCA [6] LDA [7, 8], LPP [26], NFLS [31] and UDNFLA [34] schemes. The NN classifier is applied for final classification. Note that, PCA as a preprocessing step is applied for LDA, LPP, UDNFLA and FLNFL to avoid the singularity problem. We retained 98 percent spectral energy to account for the raw data in all experiments.

The following experiments were implemented in Matlab 7.7.0. CPU: Intel Core(TM) 2.0GHz, RAM: 2G.

4.2 Parameter Selection

In Eq. (21), two parameters, \( t \) and \( K \), are involved in the mapping function. Note that the former is the parameter involved in the widely used heat kernel function. It can be determined empirically. We set \( t = 0.5 \) in the following experiments. The latter, which is the number of neighbors, decides the graph modeling, as well as the class mean vectors and weighting functions. Usually, the best choices of \( K \) for the top recognition accuracy depend on the data distribution and are not known before test. To determine the optimal value of \( K \), the leave-one-out strategy was carried out on training sets of both databases for each case under the 10-dimensional transformed subspace. Finally, it is calculated as the median of 10 runs.

Experimental results are tabulated in Fig. 3, where \( K \) ranges from 5 to 100 at the intervals of 5. We can observe that the best performance is achieved when \( K \) is set at [15, 25] and [5, 15] on the JAFFE and the Cohn-Kanade databases, respectively. In the following experiments, we set \( K = 20 \) and \( K = 5 \) on the JAFFE and the Cohn-Kanade databases respectively to test the proposed FLNFL method.
4.3 Experimental Results and Discussions

In this section, we will report the experimental results of PCA, LDA, LPP, NFLS, UDNFLA and FLNFL algorithms. Figs. 4 and 5 show the comparison plots of recognition accuracies versus dimensions of the learned subspaces on the JAFFE and Cohn-Kanade databases for different database partition cases. Concerning the LDA method, note that, there are at most \( L - 1 \) nonzero generalized eigenvalues, where \( L \) is the number of classes in the dataset. Thus, the upper bound of the dimensionality of LDA is \( L - 1 \). Table 1 summarizes the best recognition rates obtained in the optimal subspace and the corresponding dimensionalities of each method on both databases. As can be seen, FLNFL produced the highest recognition rates out of the methods compared on both databases for each strategy. For Fig. 4, LPP performed the worst; NFLS and UDNFLA were generally comparable, both superior to PCA and LDA. For Fig. 5, NFLS and UDNFLA showed slight superiority than PCA, LDA and LPP, but they all inferior to FLNFL.

![Fig. 4. Recognition rates for JAFFE database: (a) Case 1, (b) Case 2.](image-url)

![Fig. 5. Recognition rates for Cohn-Kanade database.](image-url)
Table 1. The best recognition accuracy achieved by All Methods on the JAFFE database and the Cohn-Kanade database.

<table>
<thead>
<tr>
<th>Method</th>
<th>JAFFE</th>
<th>Cohn-Kanade</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
</tr>
<tr>
<td>PCA</td>
<td>91.43(20)</td>
<td>92.86(35)</td>
</tr>
<tr>
<td>LDA</td>
<td>91.43(5)</td>
<td>92.86(5)</td>
</tr>
<tr>
<td>LPP</td>
<td>77.14(30)</td>
<td>75.71(40)</td>
</tr>
<tr>
<td>NFLS</td>
<td>85.71(35)</td>
<td>85.71(15)</td>
</tr>
<tr>
<td>UDNFLA</td>
<td>91.43(25)</td>
<td>92.86(30)</td>
</tr>
<tr>
<td>FLNFL</td>
<td>97.14(35)</td>
<td>95.71(20)</td>
</tr>
</tbody>
</table>

The numbers in the parentheses are the selected subspace dimensions.

The series of systematic experiments on two standard facial expression databases lead to the following notable findings:

(1) In all the experiments, our proposed FLNFL method performs better than PCA, LDA, LPP, especially when the training set is relative small. In our opinion, there are two reasons. The main reason is that our approach can effectively handle small sample size problem by expanding the capacity of available samples higher recognition accuracy is thus achieved. The number of training sample applied in our experiments on JAFFE and Cohn-Kanade databases are at most 20 and 75 per class respectively. The above three traditional approaches could not sufficiently cover the variations of the testing images and cannot disclose the discriminative information on JAFFE. But for the larger subset of the Cohn-Kanade database, they could work well. That is why the two NFL-based variants, NFLS and UDNFLA exhibited parallel performance on the Cohn-Kanade database. Another suggested reason is that by resorting to fuzzy assignment technique, FLNFL can capture more variations arise from environmental factors thus can explore the intrinsic distribution of samples.

(2) By taking both local information and discriminative information of data samples into account, FLNFL outperforms NFLS and UDNFLA. This fact confirms the conclusion drawn in [21] that facial expressions reside on a lower-dimensional manifold of the original image space.

(3) The selection of $K$ lies in various properties of data, such as manifold geometry, intrinsic distribution, and sampling density. As can be seen that, the performance on JAFFE tabulates strongly corresponding to different values of $K$, whereas it is slightly sensitive to the choice of $K$ on Cohn-Kanade database. Therefore, in order to find the optimal value of $K$, experiments should be taken in advance for a particular database.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new manifold embedding extension of NFLS, called FLNFL, which couples graph embedded-projection with fuzzy set theory to model the manifold structure. Referring to the Fisher type of learning manner, FLNFL derives compact and well-separated clusters by maximizing local margin of nearby points from dif-
ferent classes and minimizing similarities of nearby points from the same class. Experimental evaluations on two facial expression databases have shown that the proposed method can match the best performing state-of-the-art methods in the literature and exceed others in facial expression recognition. Our experimental results also indicate that facial expression does reside on a lower-dimensional manifold of the original image space.

As discussed before, the parameter $K$ involved in the objective function of FLNFL is tuned in a brute-force manner which is relatively time-consuming. Also it is hard to obtain a systematic strategy to find the optimal values only from a small number of experiments described this paper, since the determination of $K$ has to with various properties of data set. So our next goal is to explore the tuning process and further find an empirical way of choosing this parameter from statistical perspective.

REFERENCES


Wei Li (李薇) received B.S. degree in Computer Science from Beijing Jiaotong University, Beijing, China, in 2008. From September 2008, she is a Master-Doctoral Program student in Institute of Information Science, Department of Computer Science, Beijing Jiaotong University. Her research interests include pattern recognition and image processing.

Qiuqi Ruan (阮秋琦) Professor and Ph.D. supervisor in the Institute of Information Science, Department of Computer Science, Beijing Jiaotong University, Beijing, China. He has published 380 papers and 4 books. His research interests include pattern recognition, computer vision, multimedia information processing and image processing.
Jun Wan (万军) received B.S. degree in the School of Geophysics and Geoinformation Systems from China University of Geosciences (Beijing), Beijing, China, in 2008. From September 2008 to August 2009, he is a Master student in School of Electronic and Information Engineering Department, Beijing Jiaotong University. From September 2009, he is a Master-Doctoral Program student in Institute of Information Science, Department of Computer Science, Beijing Jiaotong University. His research interests include pattern recognition and image processing.