WILEY | Hindawi

*Research Article*

# An Improved EMD-Based Dissimilarity Metric for Unsupervised Linear Subspace Learning

**Xiangchun Yu** (iD),[1] **Zhezhou Yu** (iD),[1] **Wei Pang,**[2] **Minghao Li,**[3] **and Lei Wu**[1]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK*
[3]*College of Software, Jilin University, Changchun 130012, China*

Correspondence should be addressed to Zhezhou Yu; yuzz@jlu.edu.cn

We investigate a novel way of robust face image feature extraction by adopting the methods based on Unsupervised Linear Subspace Learning to extract a small number of good features. Firstly, the face image is divided into blocks with the specified size, and then we propose and extract pooled Histogram of Oriented Gradient (pHOG) over each block. Secondly, an improved Earth Mover's Distance (EMD) metric is adopted to measure the dissimilarity between blocks of one face image and the corresponding blocks from the rest of face images. Thirdly, considering the limitations of the original Locality Preserving Projections (LPP), we proposed the Block Structure LPP (BSLPP), which effectively preserves the structural information of face images. Finally, an adjacency graph is constructed and a small number of good features of a face image are obtained by methods based on Unsupervised Linear Subspace Learning. A series of experiments have been conducted on several well-known face databases to evaluate the effectiveness of the proposed algorithm. In addition, we construct the noise, geometric distortion, slight translation, slight rotation AR, and Extended Yale B face databases, and we verify the robustness of the proposed algorithm when faced with a certain degree of these disturbances.

## 1. Introduction

Although many sophisticated algorithms have been proposed, face recognition is still a challenging problem affected by many external factors such as the occlusion, illumination, noise, geometric distortion, translation, and rotation of face images. Recently, face recognition algorithms based on deep learning have achieved good performance [1–6]. Stacked autoencoder (SAE) [7] is an unsupervised neural network approach, where the input and target values are the same. In SAE, the deepest hidden layer carries the features we are interested in. The input layer and the deepest hidden layer are connected by multiple encoding layers, and the deepest hidden layer and output layer are connected by multiple decoding layers. The activation values of the deepest hidden layer nodes are essentially the deep representation features which are used to perform classification tasks by feeding them to the corresponding classifier such as Softmax. In order to obtain more robust features, random noise can be added to

the input layer of SAE. This method is called Stacked Denoising Autoencoders (SDAE) [8]. In practical applications, the values of the input layer nodes can be set to be 0 with a certain probability and it can extract more robust features. However, SAE and SDAE both adopt the fully connected way to establish a connection between the input layer (hidden layer) and another hidden layer. The disadvantage is that a large number of parameters need to be learned when training SAE and SDAE. Take SDAE as an example, we set the hidden nodes to be 100, and the number of the network weights and bias will be $10^6$ when extracting the deep features over images with $100 \times 100$ pixels. In order to overcome the limitations of the fully connected network, Convolutional Neural Networks (CNN) [9, 10] are proposed, where the local connected way can effectively reduce the computational complexity of model training. In addition, another important advantage of using local connected network is that we can extract local information in the input space, which is consistent with the mechanism of the visual center. The LeNet-5

[10] model is one of the most classic CNN models. The convolution kernel is learned during model training by back propagation. Another method to learn convolution kernels is the unsupervised approach: the stacked autoencoders [7, 8] are used to extract the corresponding size of convolution kernels. In UFLDL_CNN [11], convolution kernels are learned by unsupervised Stacked Autoencoder (SAE) [7, 8] and used to perform the convolution operations at the convolutional layer. However, in addition to learning a lot of network weight values and bias, these methods need to choose several hyperparameters, for example, sparsity penalty coefficient (as in autoencoder algorithm [12, 13]) and weight penalty coefficient (as in regularization of deep neural network) [14]. Furthermore, these parameters need to be selected by cross validation, so the complexity and high computational cost are attached to these algorithms [14]. On the other hand, algorithms based on keypoints, such as sift [15] and surf [16], have demonstrated good performance in face recognition and they are robust to scale changes, rotation, illumination, and other disturbances [15–17]. But the disadvantage of these methods is that they may require a large number of keypoints, and the number of queries between keypoints is even larger. This makes it difficult to effectively perform face recognition or retrieval tasks in large-scale face image datasets.

Linear Subspace Learning is a kind of linear projection method which assumes that the high-dimensional data are located in a low-dimensional manifold that is linearly or approximately linearly embedded in the ambient space [18–22]. Linear Subspace Learning is often used in pattern recognition and computer vision tasks. Iosifidis et al. [20] proposed the optimal class representation algorithm based on the linear discriminant analysis, which increased discrimination between classes. F. Liu and X. Liu [23] proposed the Locality Enhanced Spectral Embedding (LESE) and novel Spatially Smooth Spectral Regression (SSR) methods for face recognition. It not only constructed a good locality preserving mapping but also made full use of the spatial locality information of face image matrix. Tzimiropoulos et al. [21] proposed a subspace learning algorithm based on the image gradient orientations, which has shown good performance on appearance-based object recognition. Zhang et al. [24] proposed a Linear Subspace Learning method using the sparse coding to learn a dictionary, and the aim is to fully exploit different image components. Both unsupervised and supervised criteria are proposed in order to learn the corresponding subspace. Cai et al. [22] proposed a spatially smooth subspace approach for face recognition which took full account of spatial correlation of face image and used the Laplacian penalty to learn the corresponding spatially smooth subspace. In addition, some kernel based technologies [25–28] are also applied in face recognition, and they are mainly used to explore the nonlinear relationship between face images. When nonlinear information is contained in the dataset, the kernel based techniques will exhibit good properties. These Linear Subspace Learning methods mainly fall into two categories, supervised and unsupervised methods. For supervised methods, sample labels are used in model training, provided that these labels have been manually marked. Although sometimes some supervised methods can learn good subspaces, the disadvantage is that the samples need to be marked; so in reality, unsupervised methods are more commonly used. In this research, we focus on unsupervised methods and are committed to building a good subspace by unsupervised learning algorithms. For Linear Subspace Learning, the high-dimensional input data are mapped onto a low-dimensional space by linear projection to achieve dimensionality reduction; it is a common and critical processing module in pattern recognition. Preprocessing, feature selection, feature extraction, pooling operations, and so on are implicitly or explicitly attached to dimensionality reduction operations [29]. Furthermore, the discrimination process can be viewed as a dimensionality reduction operation where high-dimensional input data are mapped onto low-dimensional class data (binary vector consisting of 0 and 1) [29].

Raw data in reality are often high-dimensional. High-dimensional data on the one hand can increase the computational burden of recognition system; on the other hand, it brings in a negative impact (arising from noise or outlier) on robust recognition tasks with limited training sample sets [29]. Importantly, raw data are often unlabeled. Therefore, this research focuses on the algorithms based on Unsupervised Linear Subspace Learning. Different from deep learning algorithms, Unsupervised Linear Subspace Learning does not need the process of selecting complex hyperparameters. Different from the keypoint ones, grids of HOG [17] or grids of pHOG over each face image are extracted and all of them are collected to form the final descriptors.

However, the raw data and the "features" extracted from them such as grids of HOG or grids of pHOG are still high-dimensional, so it is necessary to further learn the linear subspace of them. More importantly, the data from subspace can be guaranteed to have the same dimension after completing Linear Subspace Learning, and it enables subsequent classifier training, such as Softmax or SVM. In this research, in order to best evaluate the performance of the subspace learning algorithm, we adopt the nearest neighbor (NN) to be the classifier.

One of the most important tasks of Linear Subspace Learning is to construct the adjacency graph which is used to describe the nearest neighbor relationship between samples. In order to calculate the dissimilarity, a sample is drawn into a row or column which ignores the structural information of the sample. Then $L_2$ metric (Euclidean distance) is often used to measure the dissimilarity between any two samples.

Face images taken from cameras often suffer from noise, geometric distortions [30, 31] and sometimes complex geometric distortion can occur during shooting, storage, and transmission. The most common geometric distortion is radial distortion, which includes barrel distortion and pincushion distortion. Some examples of face images suffering from noise, geometric distortions, slight translation, and rotation changes are shown in Figure 1.

In this research, we consider the algorithms that are robust to a certain degree of noise, geometric distortions, slight translation, and slight rotation changes. We construct the noise, geometric distortion, slight translation, slight rotation AR, and Extended Yale B face databases, and we

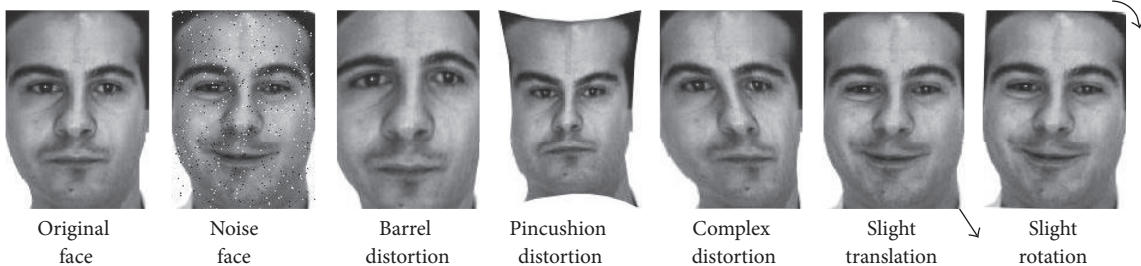| Original face | Noise face | Barrel distortion | Pincushion distortion | Complex distortion | Slight translation | Slight rotation |

FIGURE 1: Face samples suffering from noise, geometric distortions, slight translation, and rotation changes on AR face database.

also verify the robustness of our proposed algorithms to a certain degree of these disturbances. See Section 4 for detailed information about these databases.

The main contributions of our proposed algorithm are as follows.

(1) In order to reduce the computational complexity and enhance robustness, we propose and perform pooling operations on the "granularity" of cell over each block. That is to say, we accumulate the histograms for all cells over the block and we obtain a pHOG histogram over the block. Then, an improved EMD metric instead of $L_2$ metric is adopted to compare any two pHOG histograms over corresponding blocks from two different face images. It can effectively deal with the quantization problem of rigid binning.

(2) We attach great importance to the structural information of samples. In order to effectively preserve the structural information of the sample, each face image is divided into blocks with the specified size and we propose the Block Structure LPP (BSLPP) algorithm based on the improved EMD metric, which overcomes the limitation of the original LPP.

(3) We construct the noise, geometric distortion, slight translation, slight rotation AR, and Extended Yale B face databases and verify the robustness of the algorithm against a certain degree of these disturbances.

The rest of this paper is organized as follows: we first review related work in Section 2. In Section 3, we present our improved our improved EMD-based dissimilarity metric for Unsupervised Linear Subspace Learning. Experiments and results are reported in Section 4, and this is followed by the conclusions made in Section 5.

## 2. Related Work

Earth Mover's Distance (EMD) [32] is a metric proposed for some vision problems, and it can measure the dissimilarity between two distributions. EMD has been successfully applied in image retrieval, and with EMD the quantitative measure of dissimilarity between any two samples is defined by the dissimilarity of two distributions, which correlates to human perception to some extent [32].

An intuitive explanation of EMD is as follows: given two distributions (normalized histograms), one is taken as "supply" with a mass of earth properly spreading in space, and the other is regarded as "demand" with collection of holes. So, the solution is the minimal work (cost) that must be done to

fill the holes with earth [32]. And the formula of EMD defined by Rubner et al. [32] is given as follows:

$$\text{EMD} = \min_{f_{ij}} \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}} \tag{1}$$

subject to

$$
\begin{aligned}
&\sum_j f_{ij} \le P_i, \\
&\sum_i f_{ij} \le Q_j, \\
&\sum_{ij} f_{ij} = \min\left(\sum_i P_i, \sum_j Q_j\right), \\
&f_{ij} \ge 0.
\end{aligned}
\tag{2}
$$

The variables involved in Formula (2) are consistent with the ones in Formula (3). Compared with the $L_2$ metric histogram matching technique, EMD (the Cross-Bin Dissimilarity Measure, as shown in Figure 2(b)) can not only effectively deal with the quantization problem of rigid binning (the Bin-by-Bin Dissimilarity Measure, as shown in Figure 2(a)), but also demonstrate robustness to shape deformation.

We explain the results of the dissimilarity measure in Figure 2: in Figure 2(a), the $L_1$ or $L_2$ metric is adopted to measure the dissimilarity. For simplicity and intuitive display, we choose the $L_1$ metric and let $D$ denote the distance between $P$ and $Q$, so $D(P, Q) = |P - Q| = 2$; in Figure 2(b), EMD is adopted to measure the dissimilarity, so $\text{EMD}(P, Q) = 1$ (according to Formula (1)). So it is not difficult to see that the "Cross-Bin Dissimilarity Measure" can effectively deal with the quantization problem of rigid binning and correlates to human perception.

However, the EMD metric can only be used for normalized histograms. More importantly, it will suffer from high computational burden, and the worst-case complexity of time for this algorithm is exponential [33]. In order to avoid the limitation of EMD, Pele and Werman proposed the EMD variant [33]: an improved EMD-based dissimilarity measure with thresholded ground distance. It is a metric for nonnormalized histograms and shows robustness to quantization, shape deformation, and occlusion. Furthermore, it is a linear

(a) The Bin-by-Bin Dissimilarity Measure

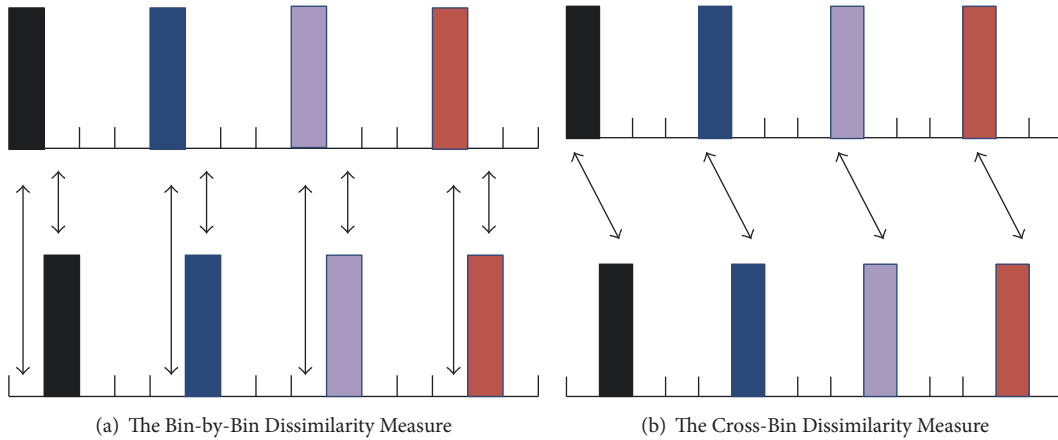(b) The Cross-Bin Dissimilarity Measure

FIGURE 2: There are 4 different colors of rectangle for each histogram, and the value of each rectangle is 0.25. The dissimilarity in (a) is 2, while the dissimilarity in (b) is 1. When $P$ denotes the histogram above, and $Q$ presents the histogram below. $P = [0.25, 0, 0, 0, 0.25, 0, 0, 0, 0.25, 0, 0, 0, 0.25, 0, 0]$, $Q = [0, 0.25, 0, 0, 0, 0.25, 0, 0, 0, 0.25, 0, 0, 0, 0.25, 0]$.

time algorithm and the time complexity is $O(N)$ [33]. Pele and Werman's EMD variant is given as follows [33]:

$$\widehat{\text{EMD}}_\alpha = \left( \min_{f_{ij}} \sum_{i,j} f_{ij} d_{ij} \right) + \left| \sum_i P_i - \sum_j Q_j \right| \\ \times \alpha \max_{i,j} \{d_{ij}\} \quad \text{s.t. (2)},$$

(3)

where $P$ and $Q$ are the two (nonnormalized) histograms and $\{f_{ij}\}$ is the flow, with each $f_{ij}$ denoting the amount of mass flowing from the $i$-th "supply" to the $j$-th "demand". $d_{ij}$ represents the thresholded ground distance, which is set to be zero for corresponding bins, one for the adjacent bins and two for other bins including the extra mass in the histogram [33]. Thresholded ground distance is just the thresholded module $L_1$ metric, and see [33] for detailed definitions. Parameter $\alpha$ in Formula (3) controls the value of the second item when the masses of $P$ and $Q$ are not equal.

The improved EMD metric is a metric for nonnormalized histograms. So, in order to measure the dissimilarity between two images by the improved EMD algorithm, we first obtain the corresponding histograms of the image and an optional one is the Histogram of Oriented Gradient (HOG). The HOG features [17] possess a certain degree of invariance to local geometric and photometric deformations, and the local shape of objects in an image can be characterized by capturing edge or gradient structure [17]. Dalal and Triggs [17] applied the HOG descriptors to human detection, which performed much better than other feature sets. They also explored the influence of the fine-scale gradients, orientation binning, spatial binning, the local contrast normalization operation, and so on, and they finally obtained the HOG descriptors for the robust visual object recognition. Zhu et al. [34] adopted a cascade of histograms of oriented gradients for fast human detection. They used the AdaBoost algorithm to select the best blocks and then built the rejector-based cascade, which not only is a near real-time human detection method but also performs well in terms of accuracy. Freeman and Roth

[35] presented the histograms of local orientation for hand gestures recognition. Newell and Griffin [36] extended the HOG and proposed multiscale histogram of oriented gradient descriptors for robust character recognition. Monzo et al. [37] compared the novel face recognition algorithm HOG-EBGM with GABOR-EBGM. The experiments showed that HOG-EBGM was more robust to illumination and rotation of images. Dniz et al. [38] employed the HOG features for face recognition. They firstly normalized the face images and then acquired the HOG descriptors using a regular grid. They also implemented a fusion strategy to combine information from different sizes of patches.

The main process of extracting the HOG features is illustrated in Figure 3.

The dimensionality of "features" is always high and contains redundant information (e.g., noise or outliers). Therefore, many features are not necessary and we aim to extract a small number of good features. Linear Subspace Learning [29, 39, 40] is one of the most powerful tools to perform dimensionality reduction. According to whether the labeled samples are used in training process, Linear Subspace Learning can be divided into three categories: the first category is Unsupervised Linear Subspace Learning [41], where no labeled samples are used; the second one is Semisupervised Linear Subspace Learning [42], where part of labeled samples are used; the last one is Supervised Linear Subspace Learning [41, 43] where all labeled samples are used.

The most typical unsupervised, semisupervised, and supervised algorithms in face recognition are Locality Preserving Projections (LPP) [18], Semisupervised Discriminant Analysis (SDA) [44], and Locality Sensitive Discriminant Analysis (LSDA) [45], respectively. However, the raw data are often unlabeled, so in this research we focus on the algorithms based on Unsupervised Linear Subspace Learning. Therefore, we adopt the typical Unsupervised Linear Subspace Learning methods LPP [18] to reduce the dimensionality of "features" of face images. More importantly, in order to make better use of the structural information of face images, we proposed a novel algorithm named Block Structure LPP (BSLPP). We
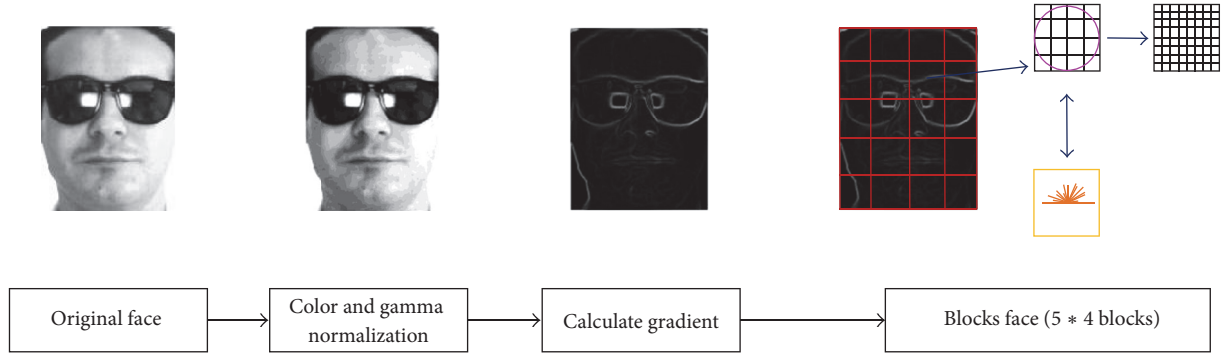
FIGURE 3: The main process of extracting the HOG features. According to [15, 17], we empirically set these parameters as follows: the block includes $4 \times 4$ cells, the size of cell is $8 \times 8$ pixels, the gradient orientation is $180°$, and the bins of histogram for $4 \times 4$ cells at the corresponding position are 12. And the final descriptor vector is made up of these 12 bins oriented gradient histograms.

also use BSLPP to reduce the dimensionality of "features" of face images.

The adjacency graph building method plays an important role in LPP and BSLPP. We adopt a dissimilarity metric based on the improved EMD metric rather than the $L_2$ metric (Euclidean metric) to conduct Unsupervised Linear Subspace Learning, where we expect to achieve better performance on the recognition rate and robustness to illumination, occlusion, noise, geometric distortion, and other disturbances.

## 3. Unsupervised Linear Subspace Learning Based on the Improved EMD

In this section, we describe our improved EMD-based dissimilarity metric for Unsupervised Linear Subspace Learning. First of all, we describe the Locality Preserving Projections (LPP) algorithm. Then, we elaborate our first algorithm (Algorithm 1): the improved EMD metric for LPP. Finally, we further introduce our second algorithm (Algorithm 2): the improved EMD metric for BSLPP.

*3.1. Locality Preserving Projections (LPP).* Locality Preserving Projections are a linear dimensionality reduction method, which falls into the graph embedding framework [46–48]. The adjacency graph building method [18, 44–48] plays an important role in the performance of LPP. The detailed steps of LPP are as follows:

(a) Use the $K$-neighborhoods to build the adjacency graph $G$, and *Node i* and *Node j* will be connected if one of the two nodes is among the $K$ nearest neighbors of the other one and the value $G_{ij}$ is set to be 1; otherwise 0.

(b) Choose the weights. The two commonly used methods are heat kernel and simple-minded, $\{0, 1\}$ [18]. We apply the $K$-nearest neighbor (KNN) to build the adjacency graph $G$ which can well present the local geometrical structure on data manifold. Let $N(x_i) = \{x_i^1, \ldots, x_i^K\}$ be the set of its $K$-nearest neighbors. We choose the simple-minded $\{0, 1\}$ weight, so the adjacency graph $G$ and the corresponding weight matrix $S$ are defined below.

$$G(i, j) \text{ or } S(i, j)$$

$$= \begin{cases} 1, & \text{if } x_i \in N(x_j) \text{ or } x_j \in N(x_i) \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

(c) Compute the projection. We solve the following generalized eigenvector problem to get the eigenvectors in accordance with the eigenvalues.

$$XLX^T a = \lambda XDX^T a, \quad (5)$$

where $X = \{x_1, x_2, \ldots, x_i, x_{i+1}, \ldots, x_n\}$ and $n$ denotes the number of samples; $L = D - S$, and $D$ is a diagonal matrix whose entries are the row or column sum of the sparse symmetric weight matrix $S$ [18], that is,

$$D(i, i) = \sum_j S(i, j). \quad (6)$$

And $a$ are the eigenvectors with respect to eigenvalues ($\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{d-1} \leq \lambda_d$).

(d) LPP embedding: $A = [a_1, a_2, \ldots, a_d]$ is the transformation matrix, and the original samples can be embedded into the $d$ dimensional subspace through the following embedding:

$$x \longrightarrow y = A^T x. \quad (7)$$

*3.2. The Improved EMD Metric for LPP.* The original Linear Subspace Learning method adopts the $L_2$ metric to calculate the dissimilarity between two samples. However, the dissimilarity between two nonnormalized histograms (such as HOG histograms) by the $L_2$ metric may suffer from the quantization problem of rigid binning, while the improved EMD metric calculates the dissimilarity between two nonnormalized histograms which correlates with human perception and tolerates the problems of quantization, distortion, occlusion, and other disturbances. The improved EMD metric has been briefly introduced in Section 2. In order to preserve the structural information of samples, we divided each face image into blocks with the specified size and extracted Histogram

**Input**: the sample set $X$ with $n$ samples, parameter $\alpha$, block parameter $M$, pHOG bins, nearest neighbors parameter $K$
**Output**: adjacency graph $G^{\text{IE}}$, weight matrix $S^{\text{IE}}$, transformation matrix $A$, eigenvalues $\lambda$, and subspace $y$
(1) **While** $i < n$
(2)     Extract HOG histogram over each block of per face image $x_i$
(3)     Carry out the pooling operation over each block and then get the pHOG histogram
(4)     Obtain the grids of pHOG vector $P$ for one face image $x_i$ and grids of pHOG vectors $\{Q\}$ for the rest of face images $\{x_j\}$, $j \neq i$
(5)     Compute the dissimilarity between $P$ and $\{Q\}$ by Equations (8) and (9)
(6)     Obtain the $K$ nearest neighbors of the face image $x_i$: $N^{\text{IE}}(x_i) = \{x_i^1, \ldots, x_i^K\}$
(7) **EndWhile**
(8) Build the adjacency graph $G^{\text{IE}}$ and calculate the corresponding weight matrix $S^{\text{IE}}$ by Equation (10)
(9) **Begin // compute the projection**
(10)    Get the diagonal matrix $L^{\text{IE}}$
(11)    Solve the generalized eigenvector problem of Equation (11) on the sample set $X$
(12)    Get the eigenvectors $a$ with respect to eigenvalues $\lambda$
(13) **End // compute the projection**
(14) Obtain the transformation matrix $A = [a_1, a_2, \ldots, a_d]$
(15) Obtain the subspace $\{y\}$ for the sample set $X$ by Equation (13)
(16) Perform face recognition by the $NN$ classifier

ALGORITHM 1: The improved EMD metric for LPP.

**Input**: the sample set $X$ with $n$ samples, parameter $\alpha$, sub adjacency graph weight parameter $\{w_m\}$, block $M$, pHOG bins, parameter nearest neighbors parameter $K$
**Output**: adjacency graph $G^{\text{IE}}$, weight matrix $S^{\text{IE}}$, transformation matrix $A$, eigenvalues $\lambda$, and subspace $y$
(1) **While** $m < M$
(2) **While** $i < n$
(3)     Extract HOG histogram over the $m$-th block of per face image $x_i$
(4)     Carry out the pooling operation over the $m$-th block and then get the pHOG histogram
(5)     Obtain pHOG vector $P_m$ over the $m$-th block for one face image $x_i$ and pHOG vectors $\{Q_m\}$ over the $m$-th blocks for the rest of face images $\{x_j\}$, $j \neq i$
(6)     Compute the dissimilarity between $P_m$ and $\{Q_m\}$ by Equations (3) and (2)
(7)     Obtain the $K$ nearest neighbors for the $m$-th block of the face image $x_i$: $N^{\text{IE}}(b_m^i) = \{b_m^1, \ldots, b_m^K\}$
(8) **EndWhile**
(9)    Obtain the adjacency graph $G^{\text{IE}}_{b_m}$ and the corresponding weight matrix $S^{\text{IE}}_{b_m}$
(10) **EndWhile**
(11) Merge these sub adjacency graphs $G^{\text{IE}}_b$ over blocks by Equation (15)
(12) Build the adjacency graph $G^{\text{IE}}$ and calculate the corresponding weight matrix $S^{\text{IE}}$ by Equation (16)
(13) **Begin // compute the projection**
(14)    Get the diagonal matrix $L^{\text{IE}}$
(15)    Solve the generalized eigenvector problem of Equation (11) on the sample set $X$
(16)    Get the the eigenvectors $a$ with respect to eigenvalues $\lambda$
(17) **End // compute the projection**
(18) Obtain the transformation matrix $A = [a_1, a_2, \ldots, a_d]$
(19) Obtain the subspace $\{y\}$ for samples set $X$ by Equation (13)
(20) Perform face recognition by the $NN$ classifier

ALGORITHM 2: The improved EMD metric for block structure LPP.

of Oriented Gradient over each block. In order to reduce the computational complexity and improve robustness, we perform the pooling operations on the "granularity" of cell over each block. And the main process of extracting the pHOG features is illustrated in Figure 4. The detailed steps of our first algorithm (Algorithm 1) are given as follows:

(a) Calculate the dissimilarity [17, 33]. The improved EMD metric is a linear time histogram metric with a low computational cost. We use this metric to calculate the

dissimilarity of the pHOG histograms (vectors) over blocks instead of the original EMD. The face image is divided into blocks and a pooled histogram of oriented gradients (pHOG) with 12 bins is obtained over each block. We compare any two pHOG histograms over corresponding blocks from two different face images by the improved EMD metric and the sum of the dissimilarity is taken as the dissimilarity between the two face images. Then we use the dissimilarity measure to obtain the $K$-nearest neighbors of each face image to build
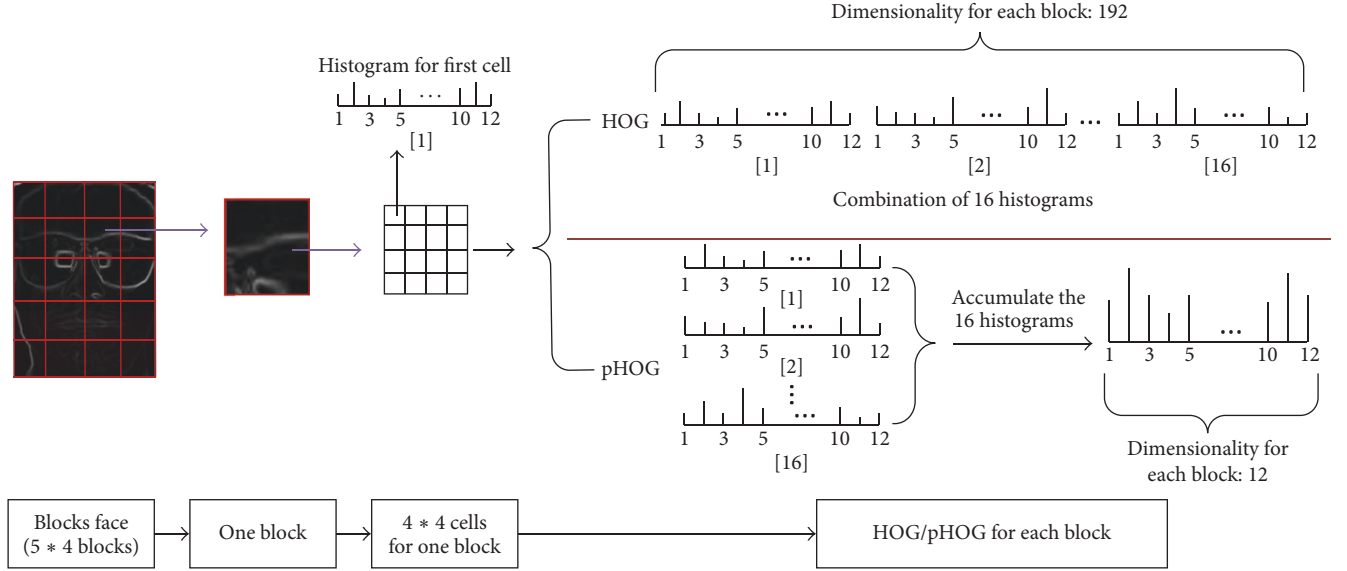
Figure 4: The main process of extracting the pHOG features.

the adjacency graph $G^{IE}$. So the final dissimilarity measure metric is as follows:

$$D(P,Q) = \sum_{m=1}^{M} \widehat{EMD}_\alpha = \sum_{m=1}^{K} \left\{ \left( \min_{f_m^{ij}} \sum_{i,j} f_m^{ij} d_m^{ij} \right) \right.$$

$$+ \left. \left| \sum_i P_m^i - \sum_j Q_m^j \right| \times \alpha \max_{i,j} \left\{ d_m^{ij} \right\} \right\}$$

(8)

subject to

$$\sum_j f_m^{ij} \le P_m^i,$$

$$\sum_i f_m^{ij} \le Q_m^j,$$

(9)

$$\sum_{ij} f_m^{ij} = \min \left( \sum_i P_m^i, \sum_j Q_m^j \right),$$

$$f_m^{ij} \ge 0.$$

In the above, $D(P,Q)$ denotes the dissimilarity of $P$ and $Q$, where $P = [P_1, P_2, \ldots, P_m, \ldots, P_M]^T$, $P_m$ denotes the pHOG histogram over the $m$-th block, similarly $Q = [Q_1, Q_2, \ldots, Q_m, \ldots, Q_M]^T$, and $Q_m$ denotes the pHOG histogram over the $m$-th block. $\{f_m^{ij}\}$ denotes the flows of the pHOG histogram over the $m$-th block: the amount transported from the $i$-th bin (supply) to the $j$-th bin (demand) is represented by $f_m^{ij}$. $d_m^{ij}$ denotes the ground distance from the $i$-th bin to the $j$-th bin. According to Pele and Werman's EMD variant [33], the $\widehat{EMD}_\alpha$ is a metric when $\alpha > 0.5$ and the ground distance $d_m^{ij}$ is also a metric. And $\alpha$ is usually set to be 1 and we also adopt the same parameter in this research. The

detailed process of calculating the dissimilarity between any two face images of the AR face database is shown in Figure 5.

(b) Chose the weights. We use the improved EMD metric (as described by (8) and (9)) to calculate the $K$-nearest neighbor. Let $N^{IE}(x_i) = \{x_i^1, \ldots, x_i^K\}$ be the set of its $K$-nearest neighbors calculated by the improved EMD metric. The adjacency graph $G^{IE}$ and the corresponding weight matrix $S^{IE}$ are defined below:

$$G^{IE}(i,j) \text{ or } S^{IE}(i,j)$$

$$= \begin{cases} 1, & \text{if } x_i \in N^{IE}(x_j) \text{ or } x_j \in N^{IE}(x_i) \\ 0, & \text{otherwise.} \end{cases}$$

(10)

(c) Compute the projection. Solve the following generalized eigenvector problem to obtain the eigenvectors in accordance with the eigenvalues.

$$XL^{IE}X^T a = \lambda X D^{IE} X^T a,$$

(11)

where $X = \{x_1, x_2, \ldots, x_i, x_{i+1}, \ldots, x_n\}$, $n$ denotes the number of training samples, $L^{IE} = D^{IE} - S^{IE}$, and $D^{IE}$ is a diagonal matrix whose entries are the row or column sum of $S^{IE}$, as shown below:

$$D^{IE}(i,i) = \sum_j S^{IE}(i,j).$$

(12)

$a$ are the eigenvectors with respect to eigenvalues ($\lambda_1 \le \lambda_2 \le \cdots \le \lambda_{d-1} \le \lambda_d$).

(d) LPP embedding: $A = [a_1, a_2, \ldots, a_d]$ is the transformation matrix, and the original samples can be embedded into a $d$ dimensional subspace through the following embedding:
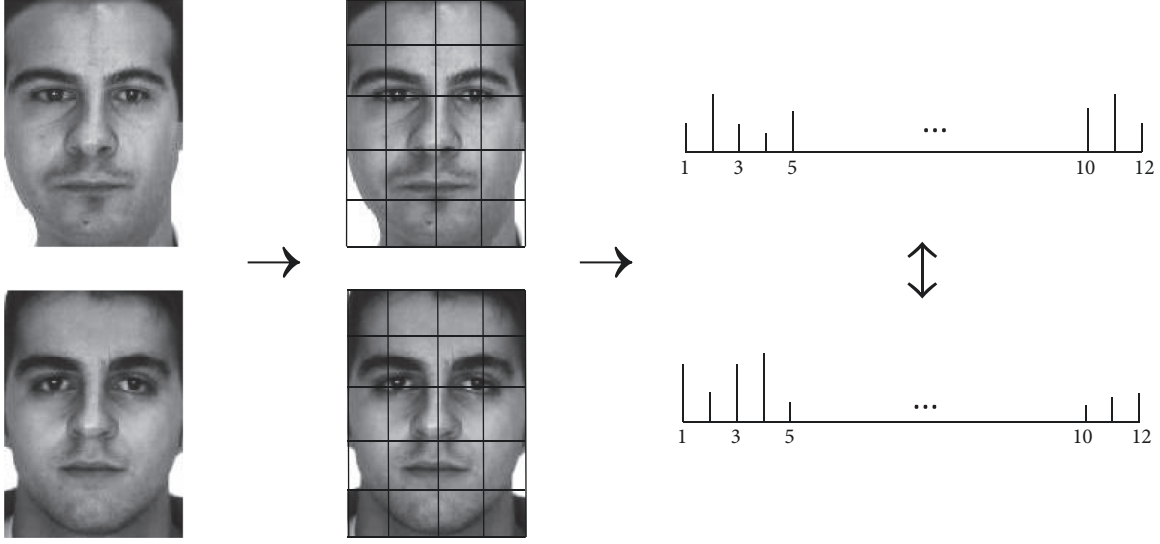
$$x \longrightarrow y = A^T x.$$

(13)

FIGURE 5: The leftmost one is the first image of the 1st and 10th persons in the AR database. The two AR face images are divided into 20 $(5 \times 4)$ blocks as shown in the middle column. The rightmost one reveals the two block histograms (in total 20 couples of histograms) between any two samples, which are used for calculating the dissimilarity (for Extended Yale B, face images are divided into 30 $(6 \times 5)$ blocks, including a total of 30 couples of histograms and the other settings are the same).

### 3.3. The Improved EMD Metric for Block Structure LPP.

In the original LPP algorithm, in order to calculate the dissimilarity, a sample is drawn into a row or column, and this ignores the structural information of the sample, which plays an important role in Linear Subspace Learning. In order to preserve the structural information of samples, we divided each face image into several blocks with the specified size. We proposed a novel algorithm named Block Structure LPP (BSLPP) based on the improved EMD metric. The main difference between Algorithms 1 and 2 is that the adjacency graph $G^{\text{IE}}$ is constructed differently. So in this section, we only elaborate the detailed process of building the adjacency graph $G^{\text{IE}}$ and other steps of Algorithm 2 are consistent with Algorithm 1.

The process of building affinity graph $G^{\text{IE}}$ in our proposed algorithm includes three main steps.

(1) We firstly calculate the dissimilarity between the block from one face image and corresponding blocks from the rest of face images with the improved EMD metric.

(2) Secondly, we get the $K$-nearest neighbors for the corresponding block and we build the sub-adjacency graph over blocks, denoted by $G_b$. Let $N^{\text{IE}}(b_m^i) = \{b_m^1, \ldots, b_m^K\}$ be the set of its $K$-nearest neighbors calculated by the improved EMD metric. Let $b^i = \{b_1^i, b_2^i, \ldots, b_m^i, \ldots, b_M^i\}$ denote all the $M$ blocks of face image $x_i$. The adjacency graph $G_{b_m}^{\text{IE}}$ and the corresponding weight matrix $S_{b_m}^{\text{IE}}$ over the $m$-th blocks of all face images are defined below.

$$
G_{b_m}^{\text{IE}}(i, j) \text{ or } S_{b_m}^{\text{IE}}(i, j)
$$
$$
= \begin{cases} 1, & \text{if } b_m^i \in N^{\text{IE}}(b_m^j) \text{ or } b_m^j \in N^{\text{IE}}(b_m^i) \\ 0, & \text{otherwise}. \end{cases} \tag{14}
$$

(3) Finally, we obtain the final adjacency graph $G^{\text{IE}}$ by merging these sub-adjacency graphs $G_b^{\text{IE}}$ over blocks. The merge function is as follows:

$$
\widetilde{G}^{\text{IE}} = \text{merge}\left(w_1 \times G_{b_1}^{\text{IE}}, w_2 \times G_{b_2}^{\text{IE}}, \ldots, w_m \right.
$$
$$
\left. \times G_{b_m}^{\text{IE}}, \ldots, w_M \times G_{b_M}^{\text{IE}}\right) = \sum_{m=1}^{M} \left(w_m \times G_{b_m}^{\text{IE}}\right). \tag{15}
$$

Among them, parameter $w_m$ denotes the weight of the $m$-th sub-adjacency graph $G_{b_m}^{\text{IE}}$. In this research, we simply set this parameter $w_m$ to be $1/M$. The final adjacency graph $G^{\text{IE}}$ and the corresponding weight matrix $S^{\text{IE}}$ are defined below.

$$
G^{\text{IE}}(i, j) \text{ or } S^{\text{IE}}(i, j) = \begin{cases} 1, & \text{if } G^{\text{IE}}(i, j) \geq \widetilde{G}^{\text{IE}}(i, :) \\ 0, & \text{otherwise}. \end{cases} \tag{16}
$$

When the final adjacency graph $G^{\text{IE}}$ and the corresponding weight matrix $S^{\text{IE}}$ are obtained, we can conduct the Block Structure LPP subspace learning. When features of face images are mapped onto a subspace, we will get the final "features" for each face image.

In this paper, we present a dissimilarity metric based on the improved EMD for Unsupervised Linear Subspace Learning. The dissimilarity between two samples is calculated by an improved EMD-based dissimilarity metric, which is a variant of the original EMD [33]. For simplicity, we refer to this dissimilarity metric as "the improved EMD metric" from now on. The whole process is described as follows.

Firstly, the $L_2$ metric will suffer from the quantization problem of rigid binning. So, the improved EMD metric [33] instead of the $L_2$ metric is adopted to compare any two pHOG histograms over the corresponding blocks from

two different face images and the sum of the dissimilarity is taken as the final dissimilarity between two different face images. The aim of the pooling operation is to reduce the computational complexity for calculating the improved EMD metric and enhance robustness against occlusion, noise, and other disturbances.

Secondly, in order to preserve the structural information of samples, each face image is divided into blocks with the specified size, and then the pHOG histogram over each block is obtained. In one way (which we call Algorithm 1), an adjacency graph is constructed by comparing $K$-nearest neighbors among face images. In another way (which we call Algorithm 2), we firstly obtain the sub-adjacency graph denoted by $G_b$ over blocks and then get the final adjacency graph $G^{IE}$ by merging these adjacency graphs over blocks.

Finally, a small number of good "features" of face images are obtained by Unsupervised Linear Subspace Learning which includes Algorithm 1 (LPP based on the improved EMD metric, named LPP_IEMD) and Algorithm 2 (BSLPP based on the improved EMD metric, named BSLPP_IEMD). When "features" of face images are mapped onto a subspace, we will get the final "features" for each face image. Among them, the "features" include the grayscale face image, grids of pHOG, and grids of HOG. See Section 4 for more detailed information about these "features."

## 4. Experiments and Results

In this section, firstly, we introduce the face databases used in this research as well as detailed experimental settings on these face databases, including training set, test set, and the choice of parameters. Secondly, we describe the experimental setups and the corresponding results for Unsupervised Linear Subspace Learning.

### 4.1. Face Databases

*4.1.1. The AR Face Database.* The AR face database has a total of 4,000 frontal images, including 126 individuals (males and females), with 26 images for each person, of which the first 13 and the last 13 were taken in two sessions (14 days). Each image has $165 \times 120$ pixels. Partial occlusions by sun glasses and scarves, illumination variation, and facial expressions occur in this database. In order to verify the effectiveness of the proposed algorithms to a certain degree of noise, geometric distortion, slight translation, and slight rotation, we construct the noise, geometric distortion, slight translation, and slight rotation AR face database.

In order to reduce the difficulty of introducing the noise, geometric distortion, slight translation, and slight rotation into the AR face database, we chose the first 15 males and the first 15 females with first 13 images (we do not consider the time factor) of each person to construct the subAR database, and this gives a total of 390 images for our experiments. We add salt and pepper noise with noise density of 0.02 to the AR face database. We use Adobe Photoshop CS6 to simulate the geometric distortions of the face images, including barrel distortion, pincushion distortion, and the complex geometric distortion. We also use Adobe Photoshop CS6 to simulate

the slight translation and slight rotation of face images. The 2nd, 6th, and 9th images of each person on our subAR face database are modified and the aim is that we consider the fusion of the simulated interference factors (noise, geometric distortion, slight translation, and slight rotation) and the inherent interference factors (occlusions, illumination, and facial expressions).

For the noise AR face database, we add salt and pepper noise with noise density of 0.02 to the 2nd, 6th, and 9th images of each person on our subAR face database.

For the geometric distortion AR face database, we add three variants, barrel distortion, pincushion distortion, and the complex geometric distortion, respectively, to the 2nd, 6th, and 9th images of each person on our subAR face database.

For the slight translation AR face database, we add slight translation to the 2nd, 6th, and 9th images of each person on our subAR face database.

For the slight rotation AR face database, we add slight rotation towards to the 2nd, 6th, and 9th images of each person on our subAR face database.

The specific details for our constructing subAR face database are shown in Figure 6.

The specific experimental settings for subAR and the noise, geometric distortion, slight translation, and slight rotation AR face databases, including training set, test set, and the choice of parameters, are as follows.

Five groups (G4/P9,…,G8/P5) of different training and testing sets are selected and we iterate every group data for 20 times, and finally we choose the average value of 20 trials as the recognition rate. G$m$/P$n$ denotes $m$ images of each person for training and $n$ images for testing, where $m + n = 13$. And the parameters of this experiment are $32 \times 32$ pixels for each block, 20 ($5 \times 4$) blocks with a length of 12 bins for each face image in total.

*4.1.2. The Extended Yale B Face Database.* The second face database used in the experiment is the Extended Yale B. The Extended Yale B face database has 2,414 face images in total, containing 38 individuals with 64 images of each person under 64 illumination conditions. Each image is in $192 \times 168$ pixels. In order to reduce the difficulty of introducing the noise, geometric distortion, slight translation, and slight rotation into the Extended Yale B face database, we chose the first 30 persons with 16 images (we choose the first one in every four of 64 face images of each person) of each person to construct the sub Extended Yale B database, and this gives a total of 480 images for our experiments. We add salt and pepper noise with noise density of 0.02 to the Extended Yale B face database. We use Photoshop to simulate the geometric distortions of the face images, including barrel distortion, pincushion distortion, and the complex geometric distortion. We also use Photoshop to simulate the slight translation and slight rotation of face images.

For the noise Extended Yale B face database, we add the noise (salt and pepper noise with noise density of 0.02) to 2nd, 6th, 10th, and 14th images of each person on our sub Extended Yale B database.
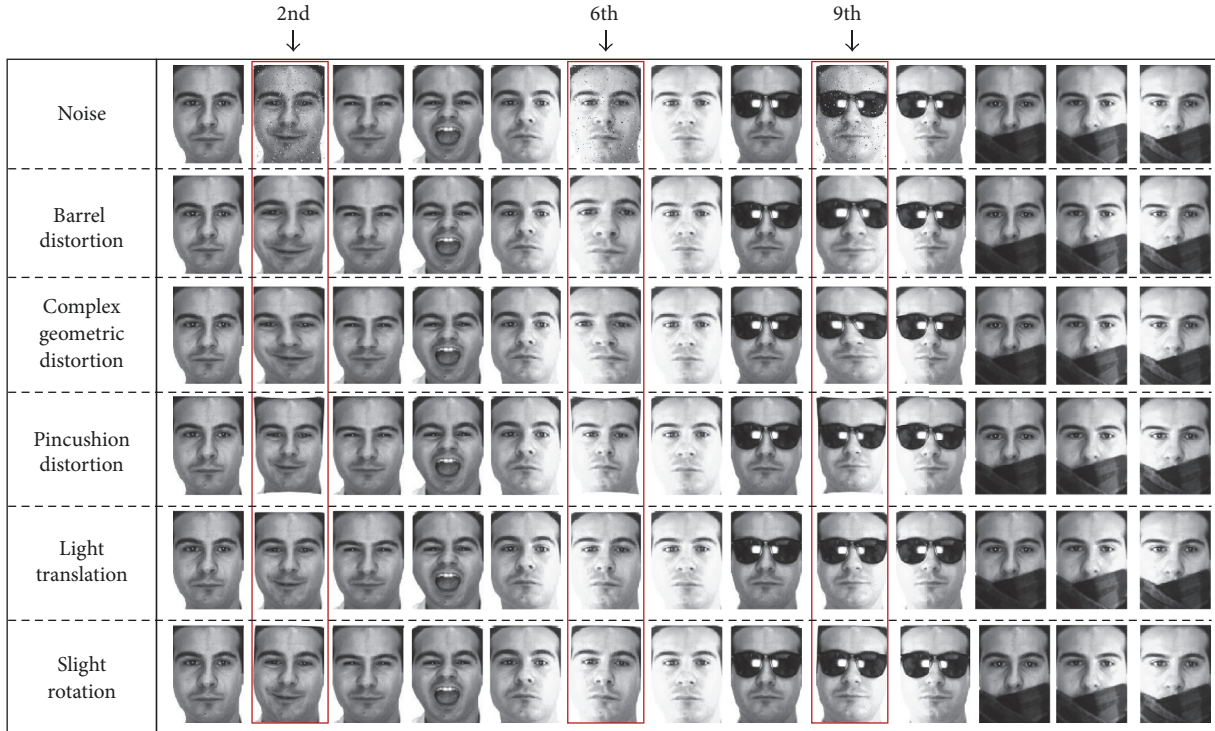
FIGURE 6: The details for the noise, geometric distortion, slight translation, and slight rotation AR face databases.

For the geometric distortion Extended Yale B face database, we add three variants, barrel distortion, pincushion distortion, and the complex geometric distortion, respectively, to the 2nd, 6th, 10th, and 14th images of each person on our sub Extended Yale B database.

For the slight translation Extended Yale B face database, we add slight translation to the 2nd, 6th, 10th, and 14th images of each person on our sub Extended Yale B database.

For the slight rotation Extended Yale B face database, we add slight rotation towards the 2nd, 6th, 10th, and 14th images of each person on our sub Extended Yale B database.

The specific details for our constructing sub Extended Yale B face database are as shown in Figure 7.

The specific experimental settings for the sub Extended Yale B and the noise, geometric distortion, slight translation, and slight rotation Extended Yale B face database, including training set, test set, and the choice of parameters, are as follows.

Five groups (G6/P10,...,G10/P6) different training and testing sets are selected and we iterate every group data for 20 times, and finally we choose the average value of 20 trials as the recognition rate. G$m$/P$n$ denotes $m$ images of each person for training and $n$ images for testing, where $m + n = 16$. And the parameters of this experiment are $32 \times 32$ pixels for each block, $30 (6 \times 5)$ blocks with a length of 12 bins for each face image in total.

### 4.2. Comparison of Experiments with Other Approaches.
Before conducting the Unsupervised Linear Subspace Learning, we conducted several experiments for comparison in order to assess the effectiveness of our algorithms. In addition to the algorithms proposed in this paper, those involved in the comparative experiments include deep learning based approaches, keypoints based approaches, and kernel based approaches. The deep learning based approaches include Stacked Denoising Autoencoders (SDAE) [7, 8, 49], LeNet-5 [10, 49], and UFLDL_CNN [11]. For SDAE and LeNet-5, we use the same settings as in [49]. For UFLDL_CNN, at the convolutional layer, 400 convolution kernels are learned by the unsupervised Stacked Autoencoder (SAE) algorithm, and then at the pooling layer we choose a pool size of 5 to conduct the pooling operation. The keypoints based approach we adopt is the sift [15] algorithm, and the specific parameters are the same as those in [15]. Kernel PCA [50] is the kernel based approach. Among them, LeNet-5 [10, 49] is the supervised algorithm, while UFLDL_CNN [11] belongs to the unsupervised one because the 400 convolution kernels are learned by SAE.

Firstly, we conducted several comparative experiments to assess the effectiveness of our algorithms on the subAR face database. We randomly selected 7 images of each face for training, and the rest for testing, and we conducted the comparative experiments on a PC with Intel(R) Core(TM) i7-4790 3.60 GHz Win 8 machine with 8 GB memory. We recorded the corresponding "cputime" (including the training and testing time) for each approach. The final results of experiments are shown in Table 1.

From Table 1, we can see that our proposed algorithm has obtained higher accuracy and consumed relatively less cputime. Although the sift approach achieves high accuracy, it consumes almost the second-longest cputime. The original face images were resized to $64 \times 64$ for UFLDL_CNN1 and
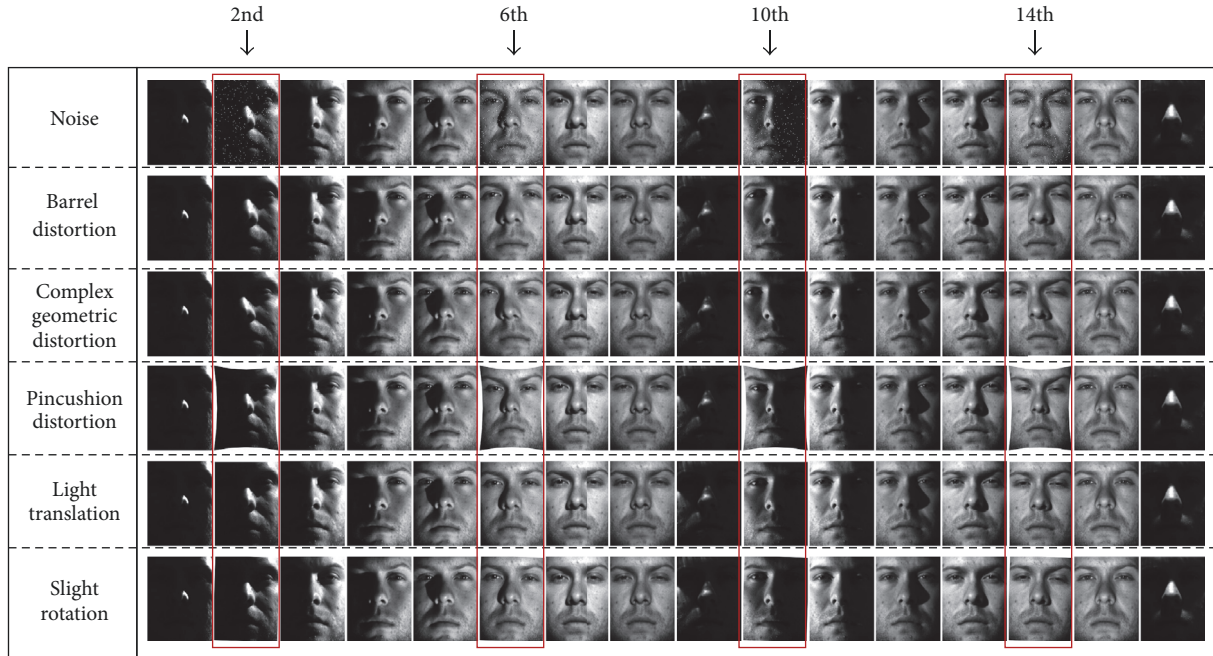
FIGURE 7: The details for the noise, geometric distortion, slight translation, and slight rotation Extended Yale B face databases.

TABLE 1: The results of comparison experiments on subAR face database.

| Algorithms | Accuracy | Cputime |
|---|---|---|
| SDAE | 0.7444 | 177 s |
| LeNet-5 | 0.5944 | 1061 s |
| sift | 0.85 | 1401 |
| UFLDL_CNN1 | 0.7055 | >437 s |
| UFLDL_CNN2 | 0.7166 | >1940 s |
| LPP | 0.4833 | 11 s |
| Kernel PCA | 0.7667 | 27 s |
| Algorithm 1 (F1) | 0.7278 | 118 s |
| Algorithm 1 (F2) | 0.8222 | 110 s |
| Algorithm 2 (F3) | 0.9444 | 59 s |

TABLE 2: The results of comparison experiments on sub Extended Yale B face database.

| Algorithms | Accuracy | Cputime (seconds) |
|---|---|---|
| SDAE | 0.5714 | 212 s |
| LeNet-5 | 0.3524 | 1327 s |
| sift | 0.4048 | 34477 s |
| UFLDL_CNN1 | 0.5857 | >533 s |
| UFLDL_CNN2 | 0.6047 | >2417 s |
| LPP | 0.6143 | 20 s |
| Kernel PCA | 0.4476 | 39 s |
| Algorithm 1 (F1) | 0.719 | 292 s |
| Algorithm 1 (F2) | 0.5619 | 274 s |
| Algorithm 2 (F3) | 0.7429 | 148 s |

resized to $128 \times 128$ for UFLDL_CNN2. The reason why the cputime is longer than 437 seconds is that we need to use SAE to learn about 400 convolution kernels and the same for UFLDL_CNN2. We also point out that our algorithms learn a subspace, which means we get a relatively small number of good features, and therefore our algorithms will spend less cputime when the unseen samples need to be tested. This is essentially an advantage of subspace learning methods over deep learning based and keypoints based ones.

Secondly, we conducted several comparative experiments to verify the effectiveness of our algorithms on the sub Extended Yale B face database. We randomly selected 7 images of each face for training, and the rest for testing. Other configurations are similar to the comparison experiments on subAR face database. The final results of comparative experiments are shown in Table 2.

From Table 2, we can see that our proposed algorithm has obtained higher accuracy with consuming relatively less cputime. However, the sift approach achieves a lower accuracy and consumes the longest cputime. LeNet-5 and Kernel PCA obtained the low accuracy and it may reveal that the supervised LeNet-5 and the kernel based kernel PCA approaches do not perform well when faced with heavy illumination variation.

*4.3. Experiments and Results on Unsupervised Linear Subspace Learning.* In this subsection, we will further demonstrate a certain degree of robustness of our proposed algorithms against partial occlusions, illumination variation, noise, geometric distortion, slight translation, and slight rotation on our constructed face databases compared with the original one.

*4.3.1. Experiments and Results on the AR Face Database.* First of all, we report the recognition rates on subAR face database.

(a) The impact of number of "bins" for our Algorithm 1 over "F2" features

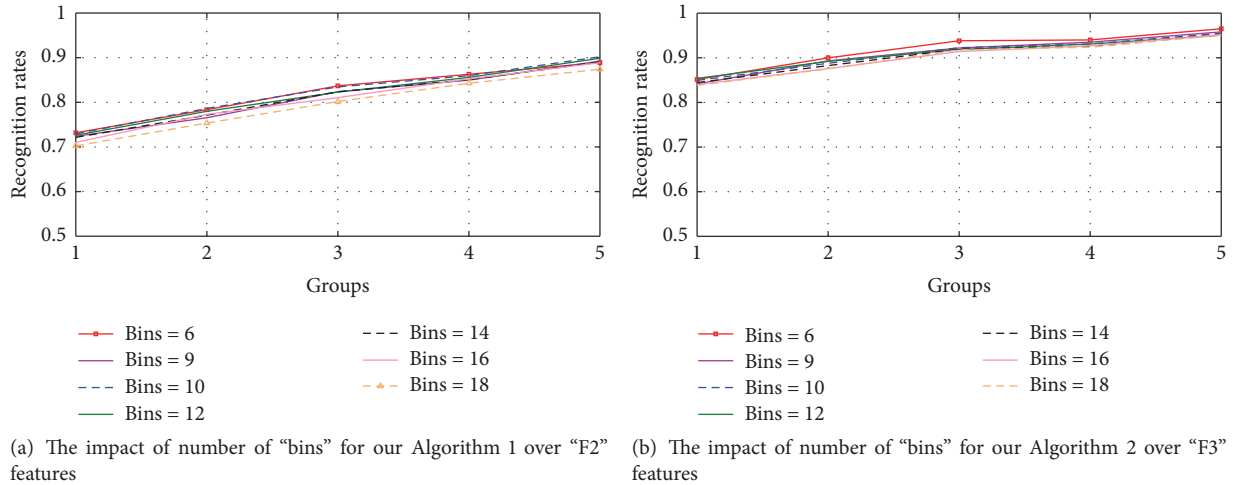(b) The impact of number of "bins" for our Algorithm 2 over "F3" features

FIGURE 8: Exploring the impact of the "bins" size on our sub AR face database.

TABLE 3: The recognition rates on subAR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|
| Baseline (F1) | 0.2213 (19800) | 0.2479 (19800) | 0.2671 (19800) | 0.2806 (19800) | 0.2867 (19800) |
| LPP (F1) | 0.4341 (75) | 0.4798 (90) | 0.5302 (110) | 0.5344 (120) | 0.5527 (135) |
| Algorithm 1 (F1)_non-pooled | 0.5402 (75) | 0.6171 (95) | 0.6767 (110) | 0.7069 (70) | 0.7360 (110) |
| Algorithm 1 (F1) | **0.6228** (75) | **0.7085** (85) | **0.7583** (100) | **0.7842** (105) | **0.8077** (105) |
| Baseline (F2) | 0.7087 (240) | 0.7735 (240) | **0.8267** (240) | 0.8439 (240) | 0.8787 (240) |
| LPP (F2) | 0.6737 (75) | 0.7423 (95) | 0.7860 (105) | 0.825 (110) | 0.8663 (120) |
| Algorithm 1 (F2)_non-pooled | 0.6530 (80) | 0.7277 (95) | 0.7800 (105) | 0.8078 (90) | 0.8480 (110) |
| Algorithm 1 (F2) | **0.7263** (80) | **0.7798** (80) | 0.8229 (100) | **0.8558** (105) | **0.8987** (120) |

In this experiment, we conduct the Unsupervised Linear Subspace Learning over the features including the grayscale face image, grids of pHOG, and grids of HOG, denoted by F1, F2, and F3, respectively. We reveal the effectiveness of the improved EMD metric, pooling HOG operation, and the BSLPP. Then, compared with the experiments on subAR face database, we obtain the experimental results on noise, geometric distortion, slight translation, and slight rotation AR face databases.

The parameter *"bins"* plays an important role in the pooling operation on the "granularity" of cell over each block, so we explore the impact of the number of "bins" on Algorithms 1 and 2 on our subAR face database. The range of the number of "bins" is (6, 9, 10, 12, 14, 16, 18) and the impact of the "bins" size on our sub AR face database is shown in Figure 8.

From Figure 8, we can see that the recognition rates with different numbers of "bins" are low. We hypothesize that it is the good performance of the improved EMD metric that leads to this result. And we selected the "bin" size of 12 in this experiment.

The recognition rates on subAR face database are shown in Table 3 and Figure 9. In Table 3, we compare three different algorithms, namely, Baseline, LPP, and Algorithm 1 (LPP_IEMD), where Baseline represents the nearest neighbor algorithm over the original "features" space. In particular, Algorithm 1 (F∗)_nonpooled means that the nonpooling

HOG with 192 bins (12 bins for the pHOG) is adopted to measure the dissimilarity between the two blocks for our Algorithm 1 over the original "F∗" features.

From Table 3 and Figure 9, we can see that our Algorithm 1 achieves the highest recognition rates over "F1" and "F2" features (except for the group of G6/P7 over "F2" features). Among them, the dimensionality is just the corresponding one of the highest recognition rate in the 20 iterations for each group. This comparison experiment verifies the effectiveness of Algorithm 1. Succinctly, we just compare three algorithms including Baseline, LPP, and our Algorithm 1 on noise, geometric distortion, slight translation, and slight rotation AR face databases.

The experimental results on noise AR face database are shown in Table 4 and Figure 10(a). As we can see from Table 4 and Figure 10(a), for the noise AR face database, our Algorithm 1 achieves the best results over "F1" features. As for "F2" features, our Algorithm 1 obtains the best results for some of the experiments. It is worth noting that although our Algorithm 1 over "F2" features does not achieve the best results, we can speed up the recognition of the unseen samples which have a smaller dimensions (good features) with a lightly lower recognition rate.

The experimental results on barrel distortion AR face database are shown in Table 5 and Figure 10(b). As we can see from Table 5 and Figure 10(b), our Algorithm 1 achieves the best results over both "F1" and "F2" features. So, it

TABLE 4: The recognition rates on noise AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Noise | Baseline (F1) | 0.2194 (19800) | 0.2394 (19800) | 0.2600 (19800) | 0.2719 (19800) | 0.2817 (19800) |
| | LPP (F1) | 0.4148 (75) | 0.4679 (100) | 0.5160 (115) | 0.5133 (140) | 0.5383 (145) |
| | Algorithm 1 (F1) | **0.5463** (65) | **0.6246** (90) | **0.6781** (120) | **0.7128** (80) | **0.7267** (90) |
| Noise | Baseline (F2) | **0.4687** (240) | 0.5323 (240) | **0.5645** (240) | **0.5864** (240) | 0.6143 (240) |
| | LPP (F2) | 0.4344 (80) | 0.4935 (95) | 0.5162 (100) | 0.5594 (110) | 0.5937 (110) |
| | Algorithm 1 (F2) | 0.4685 (85) | **0.5231** (85) | 0.5395 (95) | 0.5753 (115) | **0.6250** (125) |

TABLE 5: The recognition rates on barrel distortion AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Barrel distortion | Baseline (F1) | 0.1478 (19800) | 0.1629 (19800) | 0.1750 (19800) | 0.1794 (19800) | 0.1940 (19800) |
| | LPP (F1) | 0.2654 (75) | 0.3006 (95) | 0.3255 (115) | 0.3319 (125) | 0.3447 (135) |
| | Algorithm 1 (F1) | **0.4185** (65) | **0.4792** (80) | **0.5205** (110) | **0.5597** (120) | **0.5690** (130) |
| Barrel distortion | Baseline (F2) | 0.4974 (240) | 0.5554 (240) | 0.5921 (240) | 0.6081 (240) | 0.6400 (240) |
| | LPP (F2) | 0.4706 (80) | 0.5302 (85) | 0.5555 (105) | 0.5828 (115) | 0.6093 (120) |
| | Algorithm 1 (F2) | **0.5117** (75) | **0.5723** (90) | **0.5940** (105) | **0.6175** (115) | **0.6597** (120) |
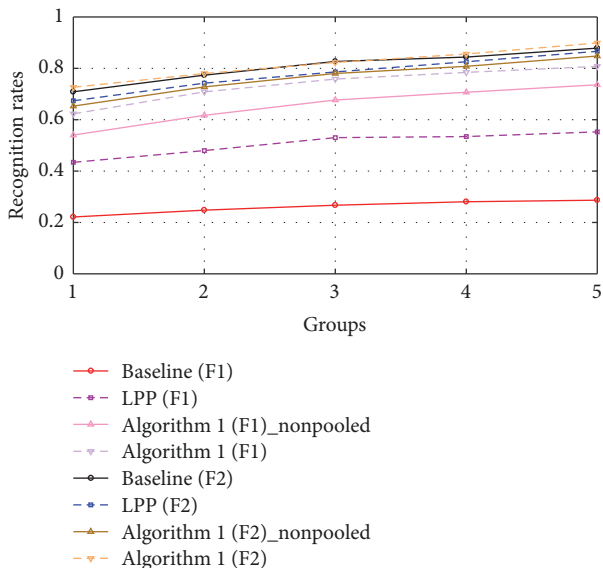


FIGURE 9: The recognition rates on subAR face database over "F1" and "F2" features.

shows that our Algorithm 1 is robust to the barrel distortion (the most common geometric distortion) to a certain degree.

The experimental results on complex geometric distortion AR face database are shown in Table 6 and Figure 10(c). As we can see from Table 6 and Figure 10(c), our Algorithm 1 achieves the best results over "F1" features. As for "F2" features, our Algorithm 1 achieves a lower recognition rate than the Baseline over "F2" features. For the complex geometric distortion, our Algorithm 1 over "F2" features may lose some discriminative information which may affect recognition rates to some extent. However, the advantage of our Algorithm 1 is that it can speed up face recognition with lower dimensionality.

The experimental results on pincushion distortion AR face database are shown in Table 7 and Figure 10(d). As we can see from Table 7 and Figure 10(d), our Algorithm 1 achieves the best results over "F1" features. As for "F2" features, our Algorithm 1 achieves a lower recognition rate than the Baseline over "F2" features. The advantage of our Algorithm 1 is that it can speed up face recognition with lower dimensionality, while the disadvantage is that our Algorithm 1 loses some discriminative information which can improve the recognition performance.

The experimental results on slight translation AR face database are shown in Table 8 and Figure 10(e). As we can see from Table 8 and Figure 10(e), our Algorithm 1 achieves the best results over both "F1" and "F2" features. So, it shows that our Algorithm 1 is robust to slight rotation to a certain degree.

The experimental results on slight rotation AR face database are shown in Table 9 and Figure 10(f). As we can see from Table 9 and Figure 10(f), our Algorithm 1 achieves the best results over both "F1" and "F2" features. So, it shows that our Algorithm 1 is robust to the slight rotation to a certain degree.

As shown in Tables 4, 6, and 7, our Algorithm 1 does not have an obvious advantage over "F2" features. "F3" features are the more robust ones, so in order to better learn the linear subspace, we adopt Algorithm 2 (BSLPP_IEMD) to conduct the Unsupervised Linear Subspace Learning over "F3" features. The experimental results on subAR are shown in Table 10 and Figure 11. In Table 10, we compare three different algorithms, namely, Baseline, LPP, and our Algorithm 2, where Baseline represents the nearest neighbor algorithm over the original "F3" features space. In particular, Algorithm 2 (F3)_$L_2$ means that $L_2$ metric is adopted to measure the dissimilarity between the two blocks for our Algorithm 2 over the original "F3" features. Algorithm 2 (F3)_nonpooled means that the nonpooling HOG with 192 bins (12 bins for the pHOG) is adopted to measure the
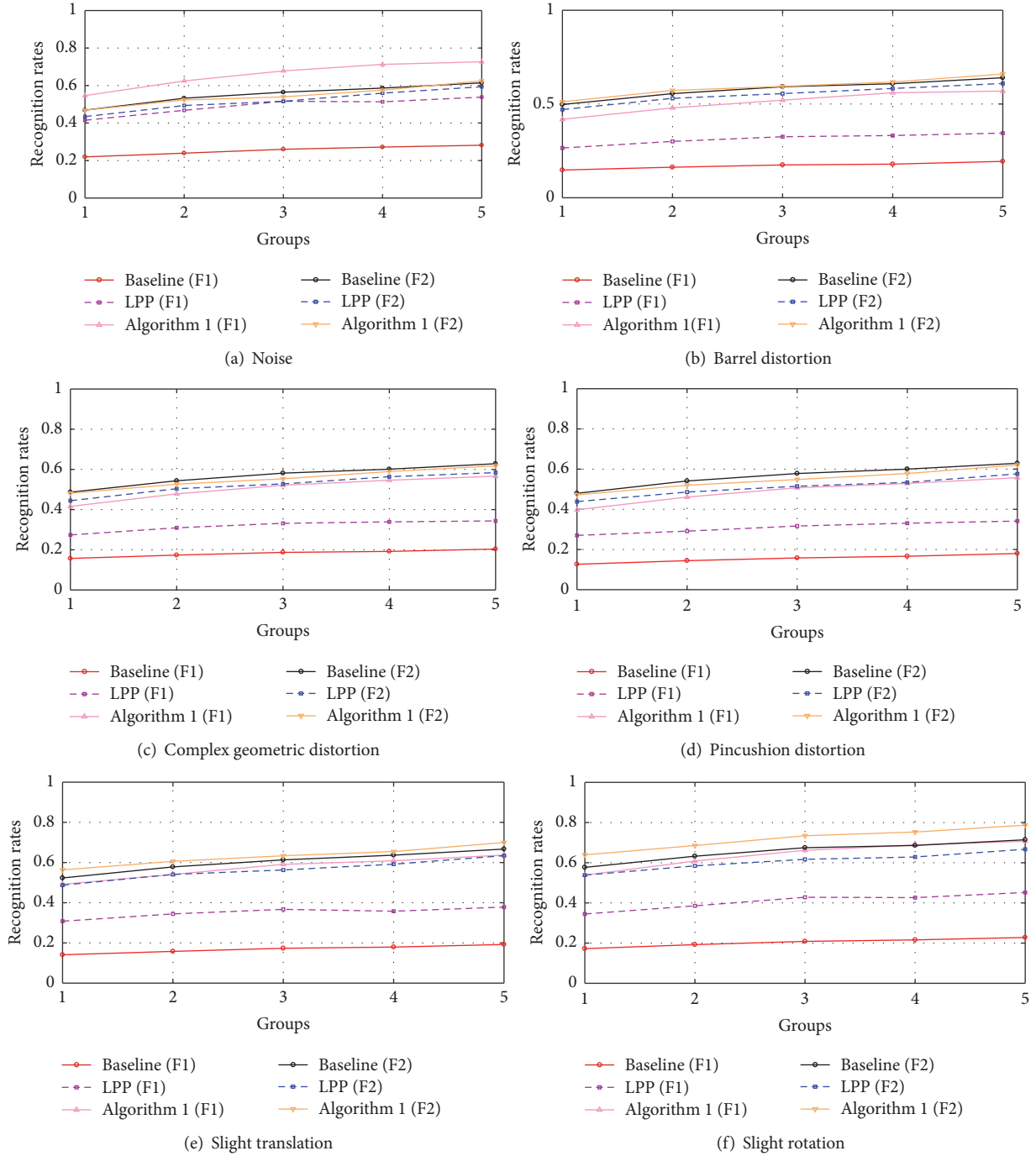
FIGURE 10: The recognition rates on noise, geometric distortion, slight translation, and slight rotation AR face databases over "F1" and "F2" features.

dissimilarity between the two blocks for our Algorithm 2 over the original "F3" features.

As one can see from Table 10 and Figure 11, Algorithm 2 achieves the highest recognition rates over "F3" features. This comparison experiment verifies the effectiveness of Algorithm 2. Succinctly, we just compare three algorithms including Baseline, LPP, Algorithm 2 (F3)_$L_2$, and Algorithm 2 on noise, geometric distortion, slight translation, and slight rotation AR face databases.

As we can see from Tables 11–15 and Figure 12, our Algorithm 2 achieves the best results over "F3" features. It shows that our Algorithm 2 is robust to the noise, geometric distortion, slight translation, and slight rotation to a certain degree. And it can well validate the effectiveness of our algorithm. More importantly, our Algorithm 2 with much lower dimensionality will provide an effective guarantee for face recognition in terms of speed and accuracy.

Table 6: The recognition rates on complex geometric distortion AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| | Baseline (F1) | 0.1570 (19800) | 0.1735 (19800) | 0.1871 (19800) | 0.1919 (19800) | 0.2037 (19800) |
| Complex geometric distortion | LPP (F1) | 0.2739 (80) | 0.3094 (95) | 0.3317 (115) | 0.3389 (130) | 0.3437 (140) |
| | Algorithm 1 (F1) | **0.4159** (80) | **0.4781** (95) | **0.5212** (115) | **0.5461** (125) | **0.5663** (115) |
| | Baseline (F2) | **0.4867** (240) | **0.5435** (240) | **0.5814** (240) | **0.6011** (240) | **0.6283** (240) |
| Complex geometric distortion | LPP (F2) | 0.4443 (80) | 0.5038 (85) | 0.5279 (105) | 0.5636 (115) | 0.5840 (105) |
| | Algorithm 1 (F2) | 0.4813 (80) | 0.5265 (95) | 0.5533 (105) | 0.5894 (115) | 0.6177 (115) |

Table 7: The recognition rates on pincushion distortion AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| | Baseline (F1) | 0.1265 (19800) | 0.1444 (19800) | 0.1579 (19800) | 0.1664 (19800) | 0.1800 (19800) |
| Pincushion distortion | LPP (F1) | 0.2702 (75) | 0.2913 (95) | 0.3169 (105) | 0.3308 (125) | 0.3413 (140) |
| | Algorithm 1 (F1) | **0.3974** (75) | **0.4610** (95) | **0.5071** (95) | **0.5286** (125) | **0.5573** (115) |
| | Baseline (F2) | **0.4796** (240) | **0.5410** (240) | **0.5781** (240) | **0.6000** (240) | **0.6290** (240) |
| Pincushion distortion | LPP (F2) | 0.4383 (80) | 0.4858 (95) | 0.5148 (105) | 0.5339 (110) | 0.5763 (115) |
| | Algorithm 1 (F2) | 0.4717 (80) | 0.5200 (95) | 0.5481 (105) | 0.5781 (95) | 0.6203 (110) |

Table 8: The recognition rates on slight translation AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| | Baseline (F1) | 0.1409 (19800) | 0.1579 (19800) | 0.1733 (19800) | 0.1794 (19800) | 0.1923 (19800) |
| Slight translation | LPP (F1) | 0.3080 (80) | 0.3444 (95) | 0.3671 (110) | 0.3578 (125) | 0.3783 (135) |
| | Algorithm 1 (F1) | **0.4917** (75) | **0.5408** (90) | **0.5910** (100) | **0.6089** (110) | **0.6373** (130) |
| | Baseline (F2) | 0.5233 (240) | 0.5783 (240) | 0.6133 (240) | 0.6367 (240) | 0.6670 (240) |
| Slight translation | LPP (F2) | 0.4876 (80) | 0.5408 (85) | 0.5631 (105) | 0.5919 (110) | 0.6343 (120) |
| | Algorithm 1 (F2) | **0.5635** (70) | **0.6060** (95) | **0.6336** (105) | **0.6547** (110) | **0.6997** (120) |

Table 9: The recognition rates on slight rotation AR face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| | Baseline (F1) | 0.1724 (19800) | 0.1925 (19800) | 0.2086 (19800) | 0.2161 (19800) | 0.2280 (19800) |
| Slight rotation | LPP (F1) | 0.3450 (80) | 0.3856 (95) | 0.4281 (115) | 0.4264 (125) | 0.4520 (140) |
| | Algorithm 1 (F1) | **0.5383** (65) | **0.6085** (95) | **0.6619** (110) | **0.6883** (125) | **0.7070** (80) |
| | Baseline (F2) | 0.5769 (240) | 0.6325 (240) | 0.6748 (240) | 0.6861 (240) | 0.7143 (240) |
| Slight rotation | LPP (F2) | 0.5385 (75) | 0.5844 (90) | 0.6171 (105) | 0.6286 (110) | 0.6670 (120) |
| | Algorithm 1 (F2) | **0.6393** (80) | **0.6856** (95) | **0.7343** (100) | **0.7528** (105) | **0.7870** (115) |

Table 10: The recognition rates on subAR face database over "F3" features; recognition rates (dimensionality).

| Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|
| Baseline (F3) | 0.7578 (3840) | 0.8254 (3840) | 0.8681 (3840) | 0.8933 (3840) | 0.9273 (3840) |
| LPP (F3) | 0.6076 (105) | 0.6596 (110) | 0.7198 (135) | 0.7583 (125) | 0.8020 (205) |
| Algorithm 2 (F3)_$L_2$ | 0.7241 (105) | 0.7846 (120) | 0.8290 (155) | 0.8633 (145) | 0.8993 (170) |
| Algorithm 2 (F3)_non-pooled | 0.7228 (105) | 0.7967 (105) | 0.8324 (150) | 0.8547 (185) | 0.8887 (150) |
| Algorithm 2 (F3) | **0.8539** (100) | **0.8919** (110) | **0.9186** (150) | **0.9308** (180) | **0.9507** (145) |

Table 11: The recognition rates on noise AR face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| | Baseline (F3) | 0.5370 (3840) | 0.6037 (3840) | 0.6326 (3840) | 0.6678 (3840) | 0.7110 (3840) |
| Noise | LPP (F3) | 0.4294 (95) | 0.4848 (115) | 0.5114 (125) | 0.5467 (165) | 0.5853 (165) |
| | Algorithm 2 (F3)_$L_2$ | 0.5133 (105) | 0.5660 (135) | 0.5955 (160) | 0.6242 (135) | 0.6663 (200) |
| | Algorithm 2 (F3) | **0.6594** (95) | **0.7102** (135) | **0.7360** (160) | **0.7525** (165) | **0.7750** (200) |

TABLE 12: The recognition rates on barrel distortion AR face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Barrel distortion | Baseline (F3) | 0.5174 (3840) | 0.5846 (3840) | 0.6179 (3840) | 0.6542 (3840) | 0.6907 (3840) |
| | LPP (F3) | 0.4102 (105) | 0.4579 (125) | 0.4967 (155) | 0.5275 (165) | 0.5597 (140) |
| | Algorithm 2 (F3)_$L_2$ | 0.5176 (100) | 0.5719 (120) | 0.6050 (160) | 0.6344 (180) | 0.6733 (175) |
| | Algorithm 2 (F3) | **0.6913** (105) | **0.7471** (125) | **0.7750** (135) | **0.8131** (180) | **0.8423** (185) |

TABLE 13: The recognition rates on complex geometric distortion AR face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Complex geometric distortion | Baseline (F3) | 0.5174 (3840) | 0.5835 (3840) | 0.6176 (3840) | 0.6556 (3840) | 0.6927 (3840) |
| | LPP (F3) | 0.4046 (105) | 0.4552 (120) | 0.4912 (125) | 0.5247 (175) | 0.5513 (175) |
| | Algorithm 2 (F3)_$L_2$ | 0.5078 (105) | 0.5750 (130) | 0.5974 (130) | 0.6281 (185) | 0.6617 (170) |
| | Algorithm 2 (F3) | **0.6837** (95) | **0.7433** (115) | **0.7748** (150) | **0.8067** (165) | **0.8287** (215) |

TABLE 14: The recognition rates on pincushion distortion AR face database over "F3" features; recognition rates (dimensionality).

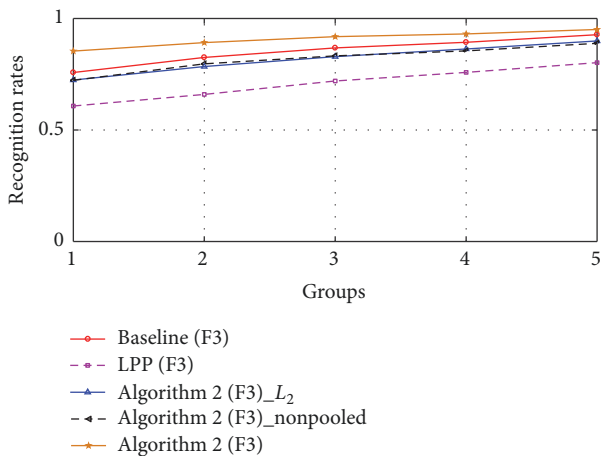| Face database | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Pincushion distortion | Baseline (F3) | 0.5352 (3840) | 0.5971 (3840) | 0.6298 (3840) | 0.6703 (3840) | 0.7090 (3840) |
| | LPP (F3) | 0.4144 (105) | 0.4594 (110) | 0.4895 (145) | 0.5225 (130) | 0.5593 (195) |
| | Algorithm 2 (F3)_$L_2$ | 0.5093 (105) | 0.5644 (105) | 0.5998 (155) | 0.6319 (145) | 0.6723 (210) |
| | Algorithm 2 (F3) | **0.6507** (100) | **0.6994** (110) | **0.7243** (160) | **0.7411** (170) | **0.7743** (205) |



FIGURE 11: The recognition rates on subAR face database over "F3" features.

The experimental results on noise AR face database over "F3" features are shown in Table 11 and Figure 12(a).

The experimental results on barrel distortion AR face database over "F3" features are shown in Table 12 and Figure 12(b).

The experimental results on barrel distortion AR face database over "F3" features are shown in Table 13 and Figure 12(c).

The experimental results on pincushion distortion AR face database over "F3" features are shown in Table 14 and Figure 12(d).

The experimental results on slight translation AR face database over "F3" features are shown in Table 15 and Figures 12(e) and 12(f).

*4.3.2. Experiments and Results on the Extended Yale B.* Similar to the experiments on AR face database, we get the experimental results on sub Extended Yale B, noise, geometric distortion, slight translation, and slight rotation Extended Yale B face databases. The recognition rates are shown in Tables 16–21 and Figure 13. In Tables 16–21, we compare three different algorithms, namely, Baseline, LPP, and our Algorithm 1.

As we can see from Tables 16–21 and Figure 13, Algorithm 1 achieves the best results over "F1" features. It is worth noting that Algorithm 1 over "F1" features is even better than that over "F2" features. As for "F2" features, Algorithm 1 achieves the partial best results on noise, complex geometric distortion, slight translation, and slight rotation Extended Yale B face databases. So our conclusion is that Algorithm 1 over "F2" features is less effective than that over "F1" features in the case of suffering from heavily varying illumination.

The recognition rates on sub Extended Yale B face database are shown in Table 16 and Figure 13(a). The recognition rates on noise Extended Yale B face database are shown in Table 17 and Figure 13(b). The recognition rates on barrel distortion Extended Yale B face database are shown in Table 18 and Figure 13(c). The recognition rates on complex geometric distortion Extended Yale B face database are shown in Table 19 and Figure 13(d). The recognition rates on pincushion distortion Extended Yale B face database are shown in Table 20 and Figure 13(e). The recognition rates on slight translation and rotation Extended Yale B face databases are shown in Table 21 and Figures 13(f) and 13(g).

In order to better deal with the problem of heavily varying illumination and solve the serious lighting problem, we adopt Algorithm 2 to conduct the Unsupervised Linear Subspace Learning over "F3" features. The experimental results on sub, noise, geometric distortion, slight translation, and slight

TABLE 15: The recognition rates on slight translation and rotation AR face databases over "F3" features; recognition rates (dimensionality).

| Face databases | Algorithms | G4/P9 | G5/P8 | G6/P7 | G7/P6 | G8/P5 |
|---|---|---|---|---|---|---|
| Slight translation | Baseline (F3) | 0.6611 (3840) | 0.7160 (3840) | 0.7655 (3840) | 0.7967 (3840) | 0.8327 (3840) |
| | LPP (F3) | 0.5174 (100) | 0.5548 (130) | 0.6033 (160) | 0.6356 (135) | 0.6807 (210) |
| | Algorithm 2 (F3)_$L_2$ | 0.6346 (100) | 0.6985 (125) | 0.7493 (155) | 0.7689 (150) | 0.8140 (200) |
| | Algorithm 2 (F3) | **0.7798** (100) | **0.8265** (130) | **0.8550** (130) | **0.8717** (185) | **0.8980** (210) |
| Slight rotation | Baseline (F3) | 0.7578 (3840) | 0.8254 (3840) | 0.8681 (3840) | 0.8933 (3840) | 0.9273 (3840) |
| | LPP (F3) | 0.6076 (105) | 0.6596 (110) | 0.7198 (135) | 0.7583 (125) | 0.8020 (205) |
| | Algorithm 2 (F3)_$L_2$ | 0.7241 (105) | 0.7846 (120) | 0.8290 (155) | 0.8633 (145) | 0.8993 (170) |
| | Algorithm 2 (F3) | **0.8441** (105) | **0.8844** (125) | **0.9124** (150) | **0.9222** (155) | **0.9433** (155) |

TABLE 16: The recognition rates on sub Extended Yale B face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| sub | Baseline (F1) | 0.2615 (32256) | 0.2770 (32256) | 0.2904 (32256) | 0.3057 (32256) | 0.3100 (32256) |
| | LPP (F1) | 0.4955 (90) | 0.5363 (95) | 0.5598 (100) | 0.5862 (105) | 0.6208 (120) |
| | Algorithm 1 (F1) | **0.6108** (90) | **0.6585** (100) | **0.6654** (105) | **0.6831** (110) | **0.7097** (115) |
| sub | Baseline (F2) | **0.4930** (360) | **0.5187** (360) | **0.5421** (360) | **0.5705** (360) | **0.5900** (360) |
| | LPP (F2) | 0.3883 (120) | 0.4139 (135) | 0.4327 (155) | 0.4490 (165) | 0.4664 (160) |
| | Algorithm 1 (F2) | 0.4578 (130) | 0.4735 (135) | 0.4940 (155) | 0.5169 (165) | 0.5331 (165) |

TABLE 17: The recognition rates on noise Extended Yale B face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Noise | Baseline (F1) | 0.2623 (32256) | 0.2750 (32256) | 0.2833 (32256) | 0.3057 (32256) | 0.3119 (32256) |
| | LPP (F1) | 0.4345 (105) | 0.4939 (115) | 0.5123 (120) | 0.5443 (135) | 0.5839 (140) |
| | Algorithm 1 (F1) | **0.5152** (105) | **0.5663** (115) | **0.5758** (130) | **0.6055** (125) | **0.6350** (95) |
| Noise | Baseline (F2) | 0.2435 (360) | 0.2496 (360) | 0.2654 (360) | **0.2821** (360) | 0.2881 (360) |
| | LPP (F2) | 0.1802 (125) | 0.1963 (140) | 0.2031 (140) | 0.2131 (150) | 0.2283 (170) |
| | Algorithm 1 (F2) | **0.2530** (95) | **0.2604** (120) | **0.2702** (145) | 0.2795 (160) | **0.2917** (165) |

TABLE 18: The recognition rates on barrel distortion Extended Yale B face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Barrel distortion | Baseline (F1) | 0.1732 (32256) | 0.1746 (32256) | 0.1835 (32256) | 0.1936 (32256) | 0.1922 (32256) |
| | LPP (F1) | 0.3282 (95) | 0.3635 (105) | 0.3860 (100) | 0.3998 (120) | 0.4336 (120) |
| | Algorithm 1 (F1) | **0.4720** (90) | **0.5024** (105) | **0.5229** (110) | **0.5479** (120) | **0.5689** (125) |
| Barrel distortion | Baseline (F2) | **0.3857** (360) | **0.3991** (360) | **0.4090** (360) | **0.4331** (360) | **0.4436** (360) |
| | LPP (F2) | 0.2970 (125) | 0.3143 (140) | 0.3312 (150) | 0.3374 (165) | 0.3417 (170) |
| | Algorithm 1 (F2) | 0.3540 (130) | 0.3619 (135) | 0.3794 (145) | 0.3900 (165) | 0.3942 (165) |

TABLE 19: The recognition rates on complex geometric distortion Extended Yale B face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Complex geometric distortion | Baseline (F1) | 0.1900 (32256) | 0.1917 (32256) | 0.2017 (32256) | 0.2076 (32256) | 0.2089 (32256) |
| | LPP (F1) | 0.3370 (95) | 0.3724 (100) | 0.3935 (115) | 0.4088 (125) | 0.4408 (130) |
| | Algorithm 1 (F1) | **0.4610** (85) | **0.4954** (100) | **0.5156** (115) | **0.5279** (120) | **0.5503** (125) |
| Complex geometric distortion | Baseline (F2) | **0.3237** (360) | 0.3293 (360) | 0.3381 (360) | **0.3543** (360) | **0.3611** (360) |
| | LPP (F2) | 0.2600 (110) | 0.2872 (140) | 0.2852 (150) | 0.2981 (165) | 0.3069 (140) |
| | Algorithm 1 (F2) | 0.3218 (125) | **0.3331** (145) | **0.3394** (155) | 0.3493 (160) | 0.3511 (170) |
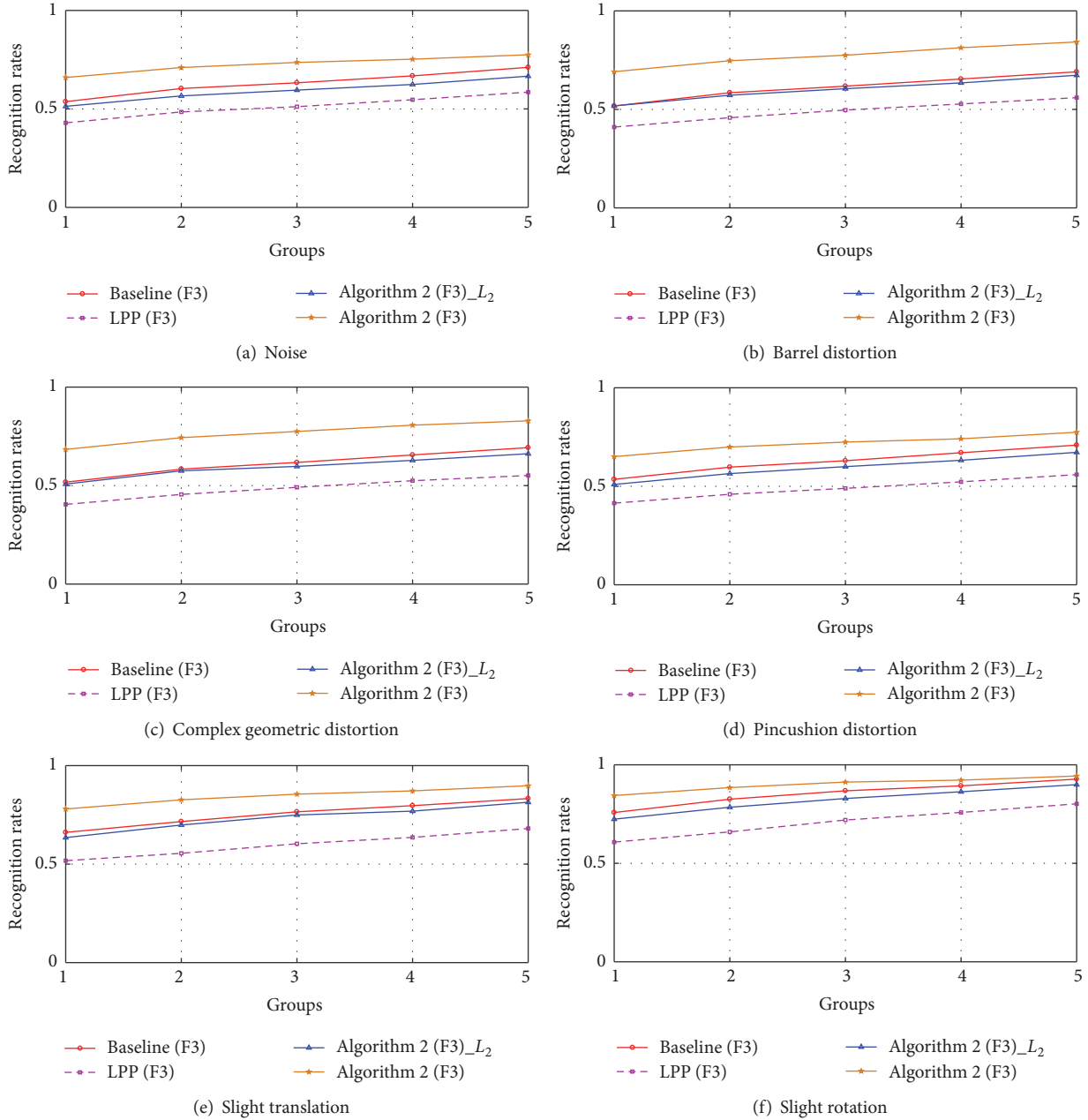
FIGURE 12: The recognition rates on noise, geometric distortion, slight translation, and slight rotation AR face databases over "F3" features.

rotation Extended Yale B face databases are shown in Tables 22–27 and Figure 14 and the specific setting of the experiment is the same as that of Tables 16–21.

As we can see from Tables 22–27 and Figures 14(a)–14(g), Algorithm 2 achieves the best results on slight translation, and slight rotation Extended Yale B over "F3" features, and the effectiveness of our algorithm against slight translation and rotation is well validated. Algorithm 2 (F3)_$L_2$ achieves the partial best results on sub, noise, complex geometric distortion, and pincushion distortion Extended Yale B face databases. The Baseline method achieves the best results on sub (except for the group of G8/P5 over "F3" features) and barrel distortion Extended Yale B. In spite of slightly lower

recognition results on sub and barrel distortion Extended Yale B, Algorithm 2 with much lower dimensionality will provide an effective guarantee for face recognition in terms of speed and accuracy.

## 5. Conclusions and Future Work

In this research, in order to reduce the computational complexity and improve robustness when calculating the improved EMD metric between numerous blocks and suffering from disturbances, we firstly carry out the pooling operation over each block to extract the pHOG features and then adopt the improved EMD metric instead of the $L_2$

(a) Sub

(b) Noise

(c) Barrel distortion

(d) Complex geometric distortion

(e) Pincushion distortion

(f) Slight translation

(g) Slight rotation

FIGURE 13: The recognition rates on sub, noise, geometric distortion, slight translation, and slight rotation Extended Yale B face databases over "F1" and "F2" features.

(a) Sub

(b) Noise

(c) Barrel distortion

(d) Complex geometric distortion

(e) Pincushion distortion

(f) Slight translation

(g) Slight rotation
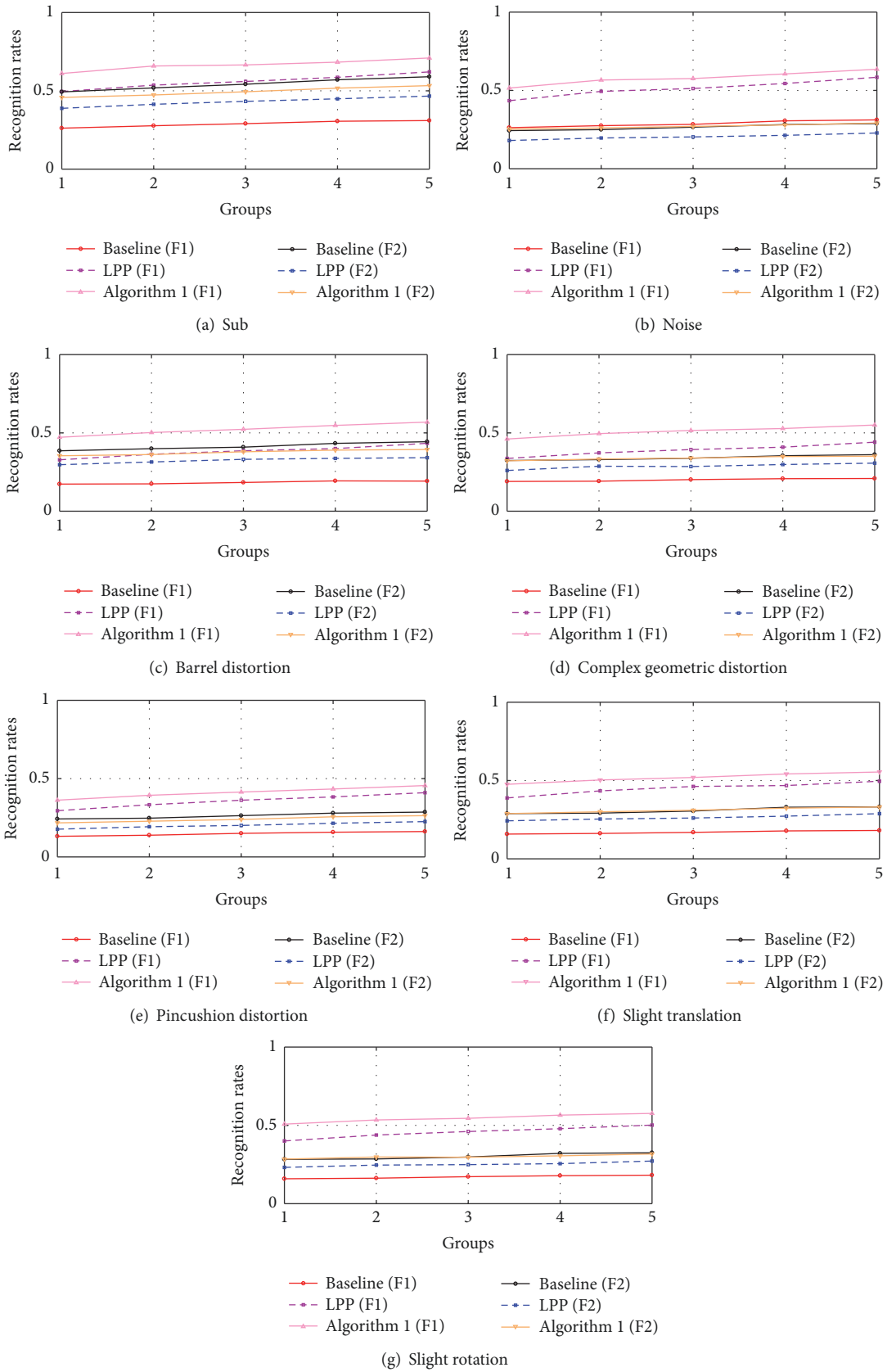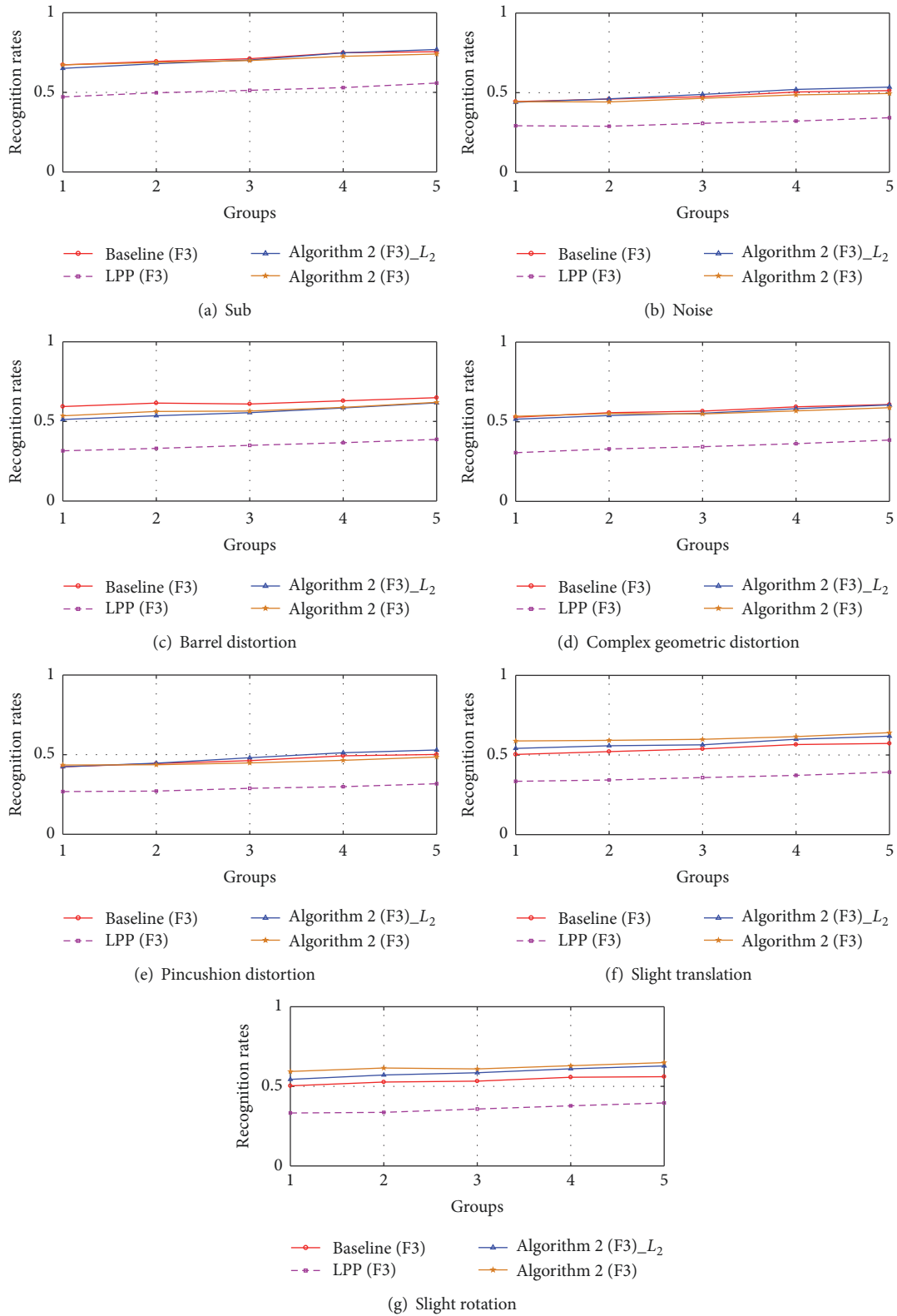
FIGURE 14: The recognition rates on noise, geometric distortion, slight translation, and slight rotation Extended Yale B face databases over "F3" features.

TABLE 20: The recognition rates on pincushion distortion Extended Yale B face database over "F1" and "F2" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Pincushion distortion | Baseline (F1) | 0.1317 (32256) | 0.1387 (32256) | 0.1506 (32256) | 0.1579 (32256) | 0.1622 (32256) |
| | LPP (F1) | 0.2952 (90) | 0.3333 (100) | 0.3623 (110) | 0.3833 (115) | 0.4103 (110) |
| | Algorithm 1 (F1) | **0.3625** (90) | **0.3937** (100) | **0.4144** (105) | **0.4338** (115) | **0.4558** (120) |
| Pincushion distortion | Baseline (F2) | **0.2425** (360) | **0.2476** (360) | **0.2637** (360) | **0.2798** (360) | **0.2864** (360) |
| | LPP (F2) | 0.1772 (120) | 0.1930 (135) | 0.2012 (145) | 0.2150 (130) | 0.2256 (160) |
| | Algorithm 1 (F2) | 0.2175 (110) | 0.2281 (120) | 0.2392 (130) | 0.2564 (155) | 0.2636 (155) |

TABLE 21: The recognition rates on slight translation and slight rotation Extended Yale B face databases over "F1" and "F2" features; recognition rates (dimensionality).

| Face databases | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Slight translation | Baseline (F1) | 0.1580 (32256) | 0.1622 (32256) | 0.1690 (32256) | 0.1783 (32256) | 0.1811 (32256) |
| | LPP (F1) | 0.3887 (90) | 0.4339 (105) | 0.4621 (105) | 0.4681 (120) | 0.4956 (120) |
| | Algorithm 1 (F1) | **0.4758** (90) | **0.5035** (95) | **0.5196** (110) | **0.5419** (120) | **0.5544** (125) |
| Slight translation | Baseline (F2) | 0.2887 (360) | 0.2920 (360) | 0.3052 (360) | **0.3295** (360) | **0.3314** (360) |
| | LPP (F2) | 0.2427 (125) | 0.2541 (140) | 0.2606 (150) | 0.2726 (160) | 0.2883 (170) |
| | Algorithm 1 (F2) | **0.2888** (125) | **0.3004** (140) | **0.3102** (150) | 0.3236 (160) | 0.3311 (160) |
| Slight rotation | Baseline (F1) | 0.1590 (32256) | 0.1628 (32256) | 0.1723 (32256) | 0.1788 (32256) | 0.1819 (32256) |
| | LPP (F1) | 0.4005 (90) | 0.4383 (105) | 0.4608 (110) | 0.4788 (120) | 0.5019 (125) |
| | Algorithm 1 (F1) | **0.5080** (90) | **0.5346** (100) | **0.5452** (105) | **0.5655** (110) | **0.5764** (120) |
| Slight rotation | Baseline (F2) | 0.2833 (360) | 0.2856 (360) | **0.2979** (360) | **0.3214** (360) | **0.3250** (360) |
| | LPP (F2) | 0.2318 (125) | 0.2465 (140) | 0.2494 (140) | 0.2557 (160) | 0.2722 (170) |
| | Algorithm 1 (F2) | **0.2848** (125) | **0.2981** (135) | 0.2967 (145) | 0.3055 (140) | 0.3175 (170) |

TABLE 22: The recognition rates on sub Extended Yale B face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| sub | Baseline (F3) | **0.6727** (5760) | **0.6946** (5760) | **0.7113** (5760) | **0.7488** (5760) | 0.7553 (5760) |
| | LPP (F3) | 0.4715 (130) | 0.4974 (145) | 0.5127 (170) | 0.5298 (120) | 0.5583 (170) |
| | Algorithm 2 (F3)_$L_2$ | 0.6507 (140) | 0.6800 (135) | 0.7031 (145) | 0.7483 (215) | **0.7692** (135) |
| | Algorithm 2 (F3) | 0.6725 (155) | 0.6880 (190) | 0.6998 (135) | 0.7262 (100) | 0.7406 (120) |

TABLE 23: The recognition rates on noise Extended Yale B face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| Noise | Baseline (F3) | **0.4442** (5760) | 0.4598 (5760) | 0.4746 (5760) | 0.5043 (5760) | 0.5131 (5760) |
| | LPP (F3) | 0.2923 (105) | 0.2893 (185) | 0.3075 (180) | 0.3217 (210) | 0.3428 (210) |
| | Algorithm 2 (F3)_$L_2$ | 0.4397 (80) | **0.4613** (115) | **0.4890** (130) | **0.5205** (210) | **0.5350** (245) |
| | Algorithm 2 (F3) | 0.4437 (150) | 0.4419 (165) | 0.4648 (215) | 0.4864 (215) | 0.4950 (235) |

metric as a dissimilarity measure to conduct Unsupervised Linear Subspace Learning, which has demonstrated a certain degree of robustness against partial occlusions, illumination variation, noise, geometric distortion, slight translation, slight rotation, and other disturbances. The experimental results on well-known databases confirm the effectiveness of our Unsupervised Linear Subspace Learning algorithms: Algorithm 1 (LPP_IEMD) and Algorithm 2 (BSLPP_IEMD).

Although our proposed algorithms achieve higher performance and demonstrate good robustness against some disturbances, there are still some limitations:

(1) Although the improved EMD metric is a linear time algorithm with $O(N)$ time complexity, compared with the $L_2$ metric, the training time of the model is longer. However, the model training process is offline, which still makes it acceptable.

(2) Heavy illumination variation is really a challenging problem. Unfortunately, our algorithms do not show a distinct advantage as they suffer from heavy illumination variation and this is an issue for future investigation.

Our future work will focus on how to more effectively measure the dissimilarity between samples, such as further

TABLE 24: The recognition rates on barrel distortion Extended Yale B face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| | Baseline (F3) | **0.5935** (5760) | **0.6152** (5760) | **0.6094** (5760) | **0.6295** (5760) | **0.6492** (5760) |
| Barrel distortion | LPP (F3) | 0.3155 (145) | 0.3306 (130) | 0.3498 (165) | 0.3664 (205) | 0.3872 (195) |
| | Algorithm 2 (F3)_$L_2$ | 0.5123 (155) | 0.5357 (145) | 0.5550 (150) | 0.5845 (130) | 0.6167 (145) |
| | Algorithm 2 (F3) | 0.5353 (115) | 0.5626 (100) | 0.5652 (95) | 0.5883 (170) | 0.6200 (150) |

TABLE 25: The recognition rates on complex geometric distortion Extended Yale B face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| | Baseline (F3) | 0.5287 (5760) | **0.5569** (5760) | **0.5669** (5760) | **0.5938** (5760) | **0.6083** (5760) |
| Complex geometric distortion | LPP (F3) | 0.3058 (145) | 0.3293 (90) | 0.3433 (170) | 0.3624 (200) | 0.3850 (230) |
| | Algorithm 2 (F3)_$L_2$ | 0.5162 (150) | 0.5391 (175) | 0.5535 (140) | 0.5821 (155) | 0.6058 (265) |
| | Algorithm 2 (F3) | **0.5342** (155) | 0.5517 (185) | 0.5492 (130) | 0.5683 (135) | 0.5883 (130) |

TABLE 26: The recognition rates on pincushion distortion Extended Yale B face database over "F3" features; recognition rates (dimensionality).

| Face database | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P65 |
|---|---|---|---|---|---|---|
| | Baseline (F3) | 0.4237 (5760) | 0.4443 (5760) | 0.4625 (5760) | 0.4929 (5760) | 0.5011 (5760) |
| Pincushion distortion | LPP (F3) | 0.2682 (120) | 0.2713 (175) | 0.2887 (180) | 0.2990 (195) | 0.3175 (210) |
| | Algorithm 2 (F3)_$L_2$ | 0.4240 (100) | **0.4465** (150) | **0.4808** (160) | **0.5121** (220) | **0.5294** (265) |
| | Algorithm 2 (F3) | **0.4350** (145) | 0.4367 (165) | 0.4481 (210) | 0.4645 (215) | 0.4856 (185) |

TABLE 27: The recognition rates on slight translation and slight rotation Extended Yale B face databases over "F3" features; recognition rates (dimensionality).

| Face databases | Algorithms | G6/P10 | G7/P9 | G8/P8 | G9/P7 | G10/P6 |
|---|---|---|---|---|---|---|
| | Baseline (F3) | 0.5035 (5760) | 0.5213 (5760) | 0.5383 (5760) | 0.5657 (5760) | 0.5725 (5760) |
| Slight translation | LPP (F3) | 0.3347 (140) | 0.3431 (155) | 0.3583 (185) | 0.3719 (205) | 0.3922 (190) |
| | Algorithm 2 (F3)_$L_2$ | 0.5415 (155) | 0.5580 (175) | 0.5640 (210) | 0.5988 (170) | 0.6183 (265) |
| | Algorithm 2 (F3) | **0.5882** (140) | **0.5920** (145) | **0.5985** (210) | **0.6155** (130) | **0.6403** (155) |
| | Baseline (F3) | 0.5033 (5760) | 0.5270 (5760) | 0.5327 (5760) | 0.5571 (5760) | 0.5606 (5760) |
| Slight rotation | LPP (F3) | 0.3327 (140) | 0.3367 (150) | 0.3577 (170) | 0.3779 (170) | 0.3958 (225) |
| | Algorithm 2 (F3)_$L_2$ | 0.5438 (160) | 0.5713 (175) | 0.5854 (205) | 0.6102 (235) | 0.6281 (260) |
| | Algorithm 2 (F3) | **0.5935** (110) | **0.6152** (155) | **0.6094** (200) | **0.6295** (110) | **0.6492** (175) |

refining the improved EMD metric and how to better represent the neighborhood relationship between samples besides KNN. To determine the weight of subadjacency graph by the adaptive weight learning method is another concern. Finally, we will also be committed to extract more robust features and make further improvement for the HOG algorithm.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

## References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[2] S. Bu, P. Han, Z. Liu, J. Han, and H. Lin, "Local deep feature learning framework for 3D shape," *Computers and Graphics*, vol. 46, pp. 117–129, 2015.

[3] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 7700, pp. 599–619, 2012.

[4] Y. Bengio, AC. Courville, and P. Vincent, "Unsupervised feature learning and deep learning," in *A review and new perspectives*, CoRR, 2012.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[6] L. Tóth and T. Grósz, "A comparison of deep neural network training methods for large vocabulary speech recognition," in

*International Conference on Text, Speech and Dialogue*, vol. 2013, pp. 36–43, Springer.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *American Association for the Advancement of Science: Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, ACM, July 2008.

[9] J. Bouvrie, "Notes on Convolutional Neural Networks," *Neural Nets*, 2006.

[10] Y. LeCun, B. Boser E, J. Denker S et al., "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, pp. 396–404, in, 1990.

[11] matlab code: https://www.amolgmahurkar.com/ classifySTLusingCNN.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[13] P. Baldi, "unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 37–49, 2012.

[14] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] H. Bay, T. Tuytelaars, and L. van Gool, "Speeded up robust features," in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, June 2005.

[18] X. He and P. Niyogi, "Locality preserving projections," *Advances in Neural Information Processing Systems*, pp. 153–160, 2004.

[19] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *Tenth IEEE International Conference on Computer Vision*, pp. 1208–1213, Beijing, China, October 2005.

[20] A. Iosifidis, A. Tefas, and I. Pitas, "On the optimal class representation in linear discriminant analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 9, pp. 1491–1497, 2013.

[21] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2454–2466, 2012.

[22] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'07*, USA, June 2007.

[23] F. Liu and X. Liu, "Locality enhanced spectral embedding and spatially smooth spectral regression for face recognition," in *Proceedings of the 2012 IEEE International Conference on Information and Automation, ICIA 2012*, pp. 299–303, China, June 2012.

[24] L. Zhang, P. Zhu, Q. Hu, and D. Zhang, "A linear subspace learning approach via sparse coding," in *Proceedings of the 2011 IEEE International Conference on Computer Vision, ICCV 2011*, pp. 755–761, November 2011.

[25] A. Iosifidis, A. Tefas, and I. Pitas, "Class-specific reference discriminant analysis with application in human behavior analysis," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 3, pp. 315–326, 2015.

[26] J. Li, W. Hao, and X. Zhang, "Learning kernel subspace for face recognition," *Neurocomputing*, vol. 151, no. 3, pp. 1187–1197, 2015.

[27] A. Iosifidis, A. Tefas, and I. Pitas, "Kernel reference discriminant analysis," *Pattern Recognition Letters*, vol. 49, pp. 85–91, 2014.

[28] S. Zafeiriou, G. Tzimiropoulos, M. Petrou, and T. Stathaki, "Regularized kernel discriminant analysis with a robust kernel for face recognition and verification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 526–534, 2012.

[29] X. Jiang, "Linear subspace learning-based dimensionality reduction," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 16–26, 2011.

[30] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[31] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[32] Y. Rubner, C. Tomasi, and LJ. Guibas, "The earth movers distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 2000.

[33] O. Pele and M. Werman, "A Linear Time Histogram Metric for Improved SIFT Matching," *Computer Vision – ECCV*, vol. 5304, pp. 495–508, 2008.

[34] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 1491–1498, IEEE, June 2006.

[35] WT. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *Proceedings of the International Workshop on Automatic Face And Gesture Recognition*, vol. 12, pp. 296–301, 1995.

[36] A. J. Newell and L. D. Griffin, "Multiscale histogram of oriented gradient descriptors for robust character recognition," in *Proceedings of the 11th International Conference on Document Analysis and Recognition, ICDAR 2011*, pp. 1085–1089, China, September 2011.

[37] D. Monzo, A. Albiol, J. Sastre, and A. Albiol, "Hog-EBGM vs. gabor-EBGM," in *Proceedings of the 2008 IEEE International Conference on Image Processing, ICIP 2008*, pp. 1636–1639, USA, October 2008.

[38] O. Dniz, G. Bueno, J. Salido, and F. D. L. Torre, "Face recognition using histograms of oriented gradients," in *Pattern Recognition Letters*, vol. 32, pp. 1598–1603, 2011.

[39] M. Yin, Y. Guo, and J. Gao, "Linear Subspace Learning via sparse dimension reduction," in *Proceedings of the 2014 International Joint Conference on Neural Networks, IJCNN 2014*, pp. 3540–3547, China, July 2014.

[40] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea, "Visual analysis of dimensionality reduction quality for parameterized projections," *Computers and Graphics*, vol. 41, no. 1, pp. 26–42, 2014.

[41] X.-Y. Jing, S. Li, D. Zhang, J. Yang, and J.-Y. Yang, "Supervised and unsupervised parallel subspace learning for large-scale image recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 10, pp. 1497–1511, 2012.

[42] D. Zhou and C. Zhang, "Semi-supervised learning using random subspace based linear embedding repulsion graph," in *Proceedings of 31st Chinese Control Conference(CCC)*, pp. 3676–3680, 2012.

[43] D. Huang, M. Storer, F. De La Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pp. 2921–2928, USA, June 2011.

[44] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–7, Rio de Janeiro, Brazil, October 2007.

[45] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality sensitive discriminant analysis," *Proceedings of IJCAI*, vol. 2007, p. 1713, 2007.

[46] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[47] M. Wan, Z. Lai, J. Shao, and Z. Jin, "Two-dimensional local graph embedding discriminant analysis (2DLGEDA) with its application to face and palm biometrics," *Neurocomputing*, vol. 73, no. 1-3, pp. 197–203, 2009.

[48] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.

[49] J. He and Y. Zhu, "Hierarchical Multi-task Learning with Application to Wafer Quality Prediction," in *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM)*, pp. 290–298, Brussels, Belgium, December 2012.

[50] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

The Scientific
World Journal

Journal of
Probability and Statistics

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Optimization

Hindawi

Submit your manuscripts at
www.hindawi.com

International Journal of
Engineering
Mathematics

International Journal of
Analysis

Journal of
Complex Analysis

Advances in
Numerical Analysis

Mathematical Problems
in Engineering

International Journal of
Differential Equations

Discrete Dynamics in
Nature and Society

International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces

Abstract and
Applied Analysis

Advances in
Mathematical Physics