Research Article

Dynamic Facial Expression Recognition Using Sparse Reserved Projection Algorithm for Low Illumination Images

Hui Li (ID)

Guangxi University of Science and Technology, Liuzhou 545006, China

Correspondence should be addressed to Hui Li; lihui@gxust.edu.cn

In this paper, a novel approach for facial expression recognition based on sparse retained projection is proposed. The locality preserving projection (LPP) algorithm is used to reduce the dimension of face image data that ensures the local near-neighbor relationship of face images. The sparse representation method is used to solve the partial occlusion of human face and the problem of light imbalance. Through sparse reconstruction, the sparse reconstruction information of expression is retained as well as the local neighborhood information of expression, which can extract more effective and judgmental internal features from the original expression data, and the obtained projection is relatively stable. The recognition results based on CK + expression database show that this method can effectively improve the facial expression recognition rate.

1. Introduction

As an important branch of pattern recognition research, face recognition technology has become a research hotspot in today's society. Face recognition technology is realized via the face image analysis and extraction of image characteristic data, such as those with the most discriminating information to identify a technology [1]. The current face recognition technology is based on a single image and image set recognition methods. In single image recognition, each image acts on behalf of a sample, whereas in the image set recognition method, a person's multiple images are used to represent a sample that constitutes an image set. Face recognition technology involves many fields such as cognitive psychology, image processing, pattern recognition, computer vision, and physiology, so there are a lot of researchers involved in face recognition research, which means face recognition technology has seen rapid development [2].

Although face recognition technology has been applied in real life, the current technology has not fully reached the mature state, and it still faces difficulties and challenges. In practical applications, the face may be affected by some natural conditions, such as the impact of light imbalance, partial shielding of the face, facial expression change, posture change, age, and many other noises, which can affect the accuracy of face recognition results. Therefore, the current research focus should be on solving these key problems that hinder the development of face recognition technology. With the attention of researchers on face recognition worldwide, face recognition technology has made great achievements. In China, the team led by researcher Shan Shiguang, as well as the team led by Professor Gao Wen and Professor Chen Xilin, made significant contributions on face recognition and detection, and they established a large-scale domestic face database and many face recognition algorithms [3]. The team led by Professors Xu Guangyou and Zhang Changshui has made significant achievements by conducting an in-depth study of face pose changes in the process of face recognition [4]. The team led by Professor Yang Jingyu has been engaged in face recognition research for a long time. The main research direction is to analyze the algebraic features of face images and propose a variety of new face recognition algorithms [5]. In addition to the research teams listed above, there are many other research teams that have been engaged in face recognition-related research for a long time and have made corresponding achievements and progress [6, 7].

The experimental face library used by authors in [8] contained sample images of 16 people, of which each person

had 9 images of faces with different shooting angles, lighting conditions, and different sizes. All the images of each person showed no significant change in expression. The recognition rate can reach up to 96% under different lighting conditions, 64% for face images of different scales, and 85% for face images of different angles. The authors in [9] proposed a method based on elastic graph matching with relatively high recognition rate, which is a method of feature information extraction based on dynamic link structure. They represented the face image as a graph structure, and the nodes of the graph were some key points in the face image (such as nose, mouth, and eyes). The links to these key information points form the edges of the graph, where each node contains a number of key information that is used to form the template of the face image. In [10], the authors used a neural network approach for face recognition. They first extracted 50 key feature information from face images and then mapped that information to the five-dimensional network space through the neural network and finally recognized and classified face images through the multilayer perception system (MLP) of the neural network. Through the continuous improvement and innovation of researchers, a variety of neural network structures have been proposed successively, and the recognition rate of face images has been improved to varying degrees.

In this paper, numerous challenges encountered in the process of face recognition were studied and analyzed. Using locality preserving projection (LPP) algorithm and sparse representation method, we found that LPP algorithm used for reducing the dimension of face image can ensure the local near-neighbor relationship of face image data. This makes the neighborhood relationship between the image samples before and after dimensionality reduction to be maintained and preserved. In addition, sparse representation can well solve partial occlusion and illumination imbalance [11], and it has strong robustness to these problems. Therefore, sparse representation was used to classify the data after dimensionality reduction. Therefore, in this paper, we combine the LPP with the classification method of sparse representation, effectively solving the impact of relevant problems on the recognition process and achieving a good recognition rate.

The rest of this paper is organized as follows. In Section 2, the local retained projection algorithm is discussed, which is based on the sparse classification. In Section 3, the proposed face dynamic expression recognition method for low illumination image is discussed. In Section 4, the experimental results are provided. Finally, the paper is concluded and future research directions are provided in Section 5.

2. Local Retained Projection Algorithm Based on Sparse Classification

In this section, first the LPP algorithm used by the proposed scheme is discussed. Next, the sparse representation classification is discussed. Both these approaches are the main approaches used by dynamic facial expression recognition techniques for low illumination images.

2.1. Locality Preserving Projection (LPP) Algorithm.
Locality preserving projection (LPP) algorithm [12] is an unsupervised dimensionality reduction algorithm, which mainly aims at dimensionality reduction of nonlinear data. Its characteristic is that it can maintain the original topological structure relationship of the data points of the sample. It projects the data from the high-dimensional manifold space to the low-dimensional space by finding a projection matrix, aiming at finding the low-dimensional linear space embedded in the high-dimensional manifold space. The LPP algorithm ensures that the original local neighbor relationship can still be maintained after dimensionality reduction by constructing the neighbor graph.

The general process of linear dimension reduction can be described as follows: For R^m given sample set $X = \{x_1, x_2, \ldots, x_k\}$ of k data in the high-dimensional space, we need to find a projection matrix A to map all the data in the high-dimensional space to a low-dimensional space R^l, where $l \ll m$, and obtain the sample point set $Y = \{y_1, y_2, \ldots, y_k\}$ corresponding to all k samples in the low-dimensional space, and we can get $y_i = Ax_i$. In this way, for each x_i, y_i can be found to make a good low-dimensional representation of it, so as to achieve the purpose of dimensionality reduction. The LPP algorithm assumes that the high-dimensional data $X = \{x_1, x_2, \ldots, x_k\}$ are embedded on the manifold plane M in the high-dimensional space R^m.

2.1.1. Building the Nearest Neighbor Graph.
LPP algorithm is the first step to build a neighbor graph G using $X = \{x_1, x_2, \ldots, x_k\}$. Consider a set of adjacent G samples, each sample point is one of the vertices; a total of k vertices, for two vertices x_i and x_j if they are as close neighbors, are connected by edges x_i and x_j, otherwise there is no boundary between x_i and x_j, to judge whether x_i and x_j are the neighbors. The relationship between them uses the following two methods:

(a) ε-domain method: by setting a threshold ε, for each sample of data x_i, calculate the Euclidian distance between it and all other samples. If $x_j \|x_i - x_j\|^2 < \varepsilon$, the sample points x_i and x_j are considered to be close neighbors, and an edge link is used between them; otherwise, they are considered to be nonclose neighbors;

(b) K-nearest neighbor method: for each data sample x_i, calculate the Euclidean distance between it and all other samples. Select the first k sample points with the smallest Euclidean distance to x_i as the nearest sample points to x_i and link them with an edge, respectively. Figure 1 vividly shows the two methods of finding the nearest sample points.

2.1.2. Determining the Weights of a Graph.
After the construction of the nearest neighbor graph in the first step, it is necessary to assign a weight to the edge of the nearest neighbor graph to construct the weight matrix W. This weight reflects the degree of similarity between nearby samples. The greater the weight, the higher the similarity
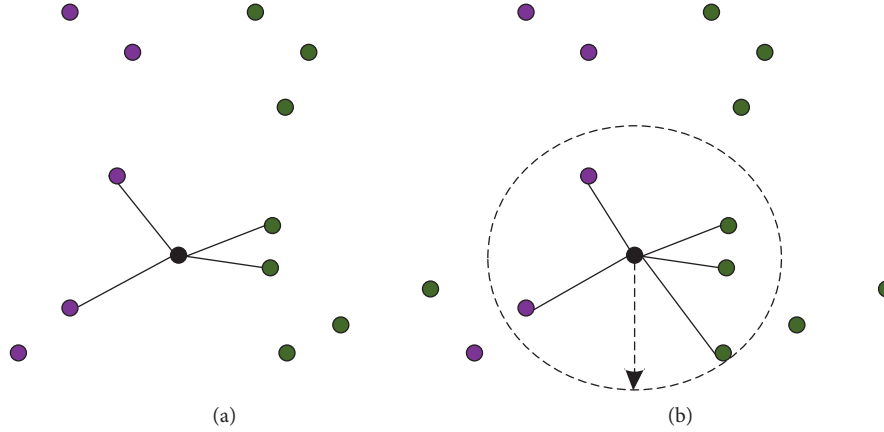
FIGURE 1: (a) k-neighborhood graph vs. (b) epsilon-neighborhood graph.

between samples. There are two methods for weight calculation:

(a) Thermonuclear-based methods:

$$
W_{ij} = \begin{cases} e^{-\left\| x_i - x_j \right\|^2 / t}, & \text{if } x_i \text{ and } x_j \text{ are adjacent,} \\ 0, & \text{if } x_i \text{ and } x_j \text{ are not adjacent.} \end{cases}
$$

(1)

(b) Method based on the 0-1 principle:

$$
W_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are adjacent,} \\ 0, & \text{if } x_i \text{ and } x_j \text{ are not adjacent.} \end{cases}
$$

(2)

2.1.3. The Target Code. The goal of the LPP algorithm is to find a projection matrix A to project data from the high-dimensional space to the low-dimensional space. The LPP algorithm obtains the projection matrix A by minimizing the following objective function:

$$
\sum_{i,j} \left(y_i - y_j \right)^2 W_{ij},
$$

(3)

where y_i and y_j are data sample points of dimensionality reduction and can be obtained by sorting out the above equation:

$$
\begin{aligned}
\frac{1}{2} \sum_{i,j} \left(y_i - y_j \right)^2 W_{ij} &= \frac{1}{2} \sum_{i,j} \left(A^T y_i - A^T y_j \right)^2 W_{ij} \\
&= \sum_{i,j} A^T x_i D_{ii} x_i^T A \\
&\quad - \sum_{i,j} A^T x_i W_{ii} x_i^T A = A^T X (D - S) X^T A \\
&= A^T X L X^T A,
\end{aligned}
$$

(4)

where $D_{ii} = \sum_{i,j} W_{ij}$ and $L = D - W$ are known as Laplacian matrices. In order to eliminate the arbitrariness and scaling

of the above minimization problem, a restriction condition is added to the above minimization problem:

$$
\sum_i D_{ii} y_j^2 = 1 \Rightarrow A^T X L X^T A - 1.
$$

(5)

With this constraint added, the above minimization problem is relatively simple and can be transformed into the minimization problem with the following constraints:

$$
\arg \min_{A^T X D X^T A = 1} A^T X L X^T A.
$$

(6)

The solution of this objective function can be regarded as the eigenvector problem of solving a generalized characteristic equation:

$$
X L X^T a = \lambda X D X^T a.
$$

(7)

Calculate all eigenvalues and eigenvectors corresponding to formula (7) and sort the eigenvalues from small to large. K eigenvalues can be obtained: $\lambda_0 < \lambda_1 < \cdots < \lambda_{k-1}$, and the eigenvector corresponding to it is $a_0, a_1, \ldots, a_{k-1}$. Take the first eigenvector to form a projection matrix $A = [a_0, a_1, \ldots, a_l]$, then

$$
x_i \longrightarrow y_i = A^T x_i,
$$

(8)

where y_j is an l-dimensional vector, which is an $m \times l$ matrix.

2.2. Sparse Representation Classification. The classification method is based on the degree of sparseness. All training samples form a data dictionary. The dictionary in each column represents a face image. Then for a new test sample, all training samples can be used to delinearize. In data analysis, the sparse coefficient vector is obtained according to the data dictionary.

2.2.1. Sparse Representation of Test Samples. Firstly, it is assumed that there are n_i face images for the training samples of class i, and each face image is represented by an m-dimensional column vector v. Then, all face samples of class i can be expressed as follows:

$A_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,n}] \in R^{m \times n_i}$, where n_i represents the number of face samples contained in class i and $v_{i,j}$ represents the j face sample of the ith individual. Then, for a new test sample y of class i, it can be linear, represented by all the training face samples of i, so y can be expressed as the following linear combination:

$$y = a_{i,1}v_{i,1} + a_{i,2}v_{i,2} + \cdots + a_{i,n_i}v_{i,n_i} \in R^m. \tag{9}$$

Here, $a_{i,j} \in R(1,2,\ldots,n_i)$ is the coefficient of a linear combination. It is assumed that y belongs to class i, because in the actual classification and recognition process, it is unknown which class y belongs to. Therefore, for all k categories (class i contains n_i training sample), a total of n training samples can be formed into a large dictionary matrix as follows:

$$A = [A_1, \ldots, A_i, A_k] = [v_{1,1}, v_{1,2}, \ldots, v_{1,n_1}, \ldots, v_{i,n_i}, \ldots, v_{k,1}, v_{k,2}, \ldots, v_{k,n_k}] \in R^{m \times n}. \tag{10}$$

Then, for any test sample y, all n training samples can be used to make a linear representation of it, that is, it can be expressed in a linear combination:

$$y = Ax \in R^m, \tag{11}$$

where x is a sparse coefficient vector. If y is a sample of class i, then a sparse coefficient vector $x = [0,0,\ldots,0,a_{i,1},a_{i,2},a_{i,n_i},0,0,\ldots,0] \in R^m$ can be obtained under ideal circumstances. It can be seen from the coefficient vector that only the coefficients corresponding to the samples belonging to the same class with y are not zero, and the other samples are all equal to zero. So, sparse coefficient vector x retained information about classes, so we can pass the sparse coefficient vectors to order a test sample y category information.

(i) Solving Sparse Solution. In the process of sparse representation, the most important problem is to solve the sparse coefficient vector x. In the practical application, we hope to find a coefficient vector x that is sparse enough and can fully represent the test sample y. For the solution of x, we can solve it by l^2 norm:

$$\hat{x}_2, \arg\min \|x\|_2, \\ s.t., Ax. \tag{12}$$

The norm problem is easy to solve by taking the pseudoinverse of the matrix A and then solving for $\hat{x}_2$. However, the vector $\hat{x}_2$ obtained by l^2 norm is not easy to test samples for classification, and the solutions obtained by it are usually relatively dense, that is, the nonzero terms in vector $\hat{x}_2$ correspond to a large number of categories, which leads to increased difficulty in classification. In order to solve the dense problem, the coefficient vector can be more strictly restricted, so the norm l^0 is used to solve:

$$\hat{x}_0 = \arg\min \|x\|_0, \\ s.t. \quad y = Ax. \tag{13}$$

The relatively sparse coefficient vector $\hat{x}_0$ can be obtained by solving the optimal solution of norm l^0 above. However, the solution of norm l^0 optimization problem has always been an NP-hard problem, and there is a problem that the solution is not unique, so its solution is not as simple as the l^2 norm. Therefore, the l^0 norm optimization solution is also

not good. Finally, a compromise method is adopted to solve the problem through the l^1 norm. Through the l^i norm, relatively sparse coefficient vectors can be obtained, and in general, there is a unique solution:

$$\hat{x}_1 = \arg\min \|x\|_1, \\ s.t. \quad y = Ax. \tag{14}$$

The desired sparse representation $\hat{x}_1$ can be obtained through the optimization solution of l^1 norm, and the following sample classification and recognition can be carried out through this coefficient vector.

(ii) Classification Based on Sparse Representation. In an ideal case, the nonzero term in the solved sparse coefficient vector $\hat{x}_1$ is the coefficient corresponding to the samples of the same class of the test sample, and the coefficient term corresponding to other samples that do not belong to the same class of the test sample is zero. Since the coefficient reflects the similarity between samples, the classification and recognition of test samples can be carried out according to the sparse coefficient vector. Samples corresponding to the term with the largest coefficient in the vector can be selected and then the test samples can be classified into the same category. In this way, classification and recognition can be realized. However, in practical applications, due to the existence of noise and model error, the situation is often not so ideal, and the nonzero term in the coefficient vector obtained is not necessarily the same category as the test sample. Therefore, the method with the largest coefficient cannot be simply selected for the classification of test samples.

For better classification, recognition, and accurate results, the concept of residual error (reconstruction error) is introduced for the sparse classification. The category to which y belongs is judged by calculating the reconstruction error of the test sample of each class. The class with the smallest reconstruction error value is regarded as the category to which y belongs. For this purpose, the following classification criterion function is used:

$$\min r_i(y) = \|y - A\delta_i(x)\|_2, \tag{15}$$

where $r_i(y) \in R$ is the reconstruction error of test sample y on class i; $\delta_i(x) \in R^n$ is a new sparse coefficient vector. It only keeps the coefficient corresponding to the item of class i

training samples in the sparse coefficient vector $\hat{x}_1$ and sets all other coefficients to zero. In this case, it is the reconstruction and recovery of test sample y on the class i. By calculating the sum of the square of the difference between the test sample y and its reconstruction recovery sample $\hat{y}_i = A\delta_i(x) \in R^m$ on class i, the reconstruction error on class i can be obtained.

3. Face Dynamic Expression Recognition Method for Low Illumination Image

First the face detection approach using AdaBoost algorithm is discussed in Section 3.1 followed by expression feature extraction based on sparse reserved projection algorithm in Section 3.2.

3.1. Face Detection Based on AdaBoost Algorithm.
The face detection algorithm based on AdaBoost algorithm consists of three parts: Haar-like feature representation, strong classifier construction, and cascade structure processing [13]. Its detection flow chart is shown in Figure 2.

3.1.1. Haar-Like Features.
There are four basic feature templates of Haar-like features, as shown in Figure 3. The template is composed of black and white rectangles, and the eigenvalue is expressed as the difference of the sum of pixels between the two rectangles. A template represents a child window used to extract local features of the face.

In Figure 3, the characteristics of class (a), (b), and (d) and the eigenvalue V are

$$v = \text{Sum}_{\text{white}} - \text{Sum}_{\text{black}}. \tag{16}$$

For the features of class (c), if the number of pixels of the black and white rectangle is the same, the calculation formula of its eigenvalue is as follows:

$$v = \text{Sum}_{\text{white}} - 2\text{Sum}_{\text{black}}. \tag{17}$$

In order to speed up the calculation speed, the face detection method based on AdaBoost algorithm calculates Haar-like eigenvalue by using integral image [14]. Save the pixel sum of the rectangle in the array to reduce duplicate operations. The integral of image I at point (x, y) and image II (x, y) is defined as

$$\text{II}(x, y) = \sum_{x' \leq x} \sum_{x' \leq x} \text{I}(x', y'), \tag{18}$$

where I (x', y') is the pixel value of image I at point (x', y'). Calculation of integral image is shown in Figure 4.

As shown in Figure 4, the integral image of each point is as follows:

$$\begin{cases} \text{II}_1 = \text{Sum}(A), \\ \text{II}_2 = \text{Sum}(A) + \text{Sum}(B), \\ \text{II}_3 = \text{Sum}(A) + \text{Sum}(C), \\ \text{II}_4 = \text{Sum}(A) + \text{Sum}(B) + \text{Sum}(C) + \text{Sum}(D). \end{cases} \tag{19}$$

Then, the sum of pixels of rectangle D is

$$\text{Sum}(D) = \text{II}_1 + \text{II}_4 - (\text{II}_2 + \text{II}_3). \tag{20}$$

It can be seen that the Haar-like eigenvalue only needs to calculate the integral image of the rectangular endpoint, which greatly reduces the calculation amount and improves the efficiency of feature extraction.

3.1.2. Strong Classifier Construction.
AdaBoost is an adaptive iterative algorithm [15]. Its basic principle is to combine multiple weak classifiers into strong classifiers. The expression for constructing the weak classifier $h_j(x)$ is as follows:

$$h_j(x) = \begin{cases} 1, & p_j f_j(x), \\ 0, & \text{otherwise}. \end{cases} \tag{21}$$

In equation (21), $f_j(x)$ represents the eigenvalue of the subwindow to be measured in the image corresponding to feature j, θ_j represents the threshold of the weak classifier, and P_j represents the direction of the inequality sign. If the positive sample is classified below θ_j, then $P_j = 1$; otherwise, $P_j = -1$.

The steps to build strong classifier $H(x)$ are as follows:

Step 1: given the training sample set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, $y_i = 1$ represents the positive sample and $y_i = 0$ represents the negative sample.

Step 2: initialize the weight of sample x_i according to equation (21):

$$w_{1/j} = \begin{cases} \dfrac{1}{2l}, & \text{if } x_i \text{ is a positive sample}, \\ \\ \dfrac{1}{2m}, & \text{if } x_i \text{ is a negative sample}. \end{cases} \tag{22}$$

In equation (22), l is the number of positive samples and m is the number of negative samples.

3.1.3. Facial Detection Based on AdaBoost Algorithm.
The last step of the method is to process the cascade structure of the generated strong classifier $H(x)$, as shown in Figure 5.

The cascading processing structure of strong classifier reflects two advantages of AdaBoost algorithm:

(1) The classifier located at the front end of the cascade structure can quickly exclude the subwindows that do not belong to the facial region. If the input image subwindow is denied once, it cannot enter the next level of classifier, thereby reducing the amount of computation.

(2) The results are more accurate because the input image subwindow is filtered layer-by-layer, only each classifier in the cascade structure is judged to belong to the face region and can finally be detected as a face.
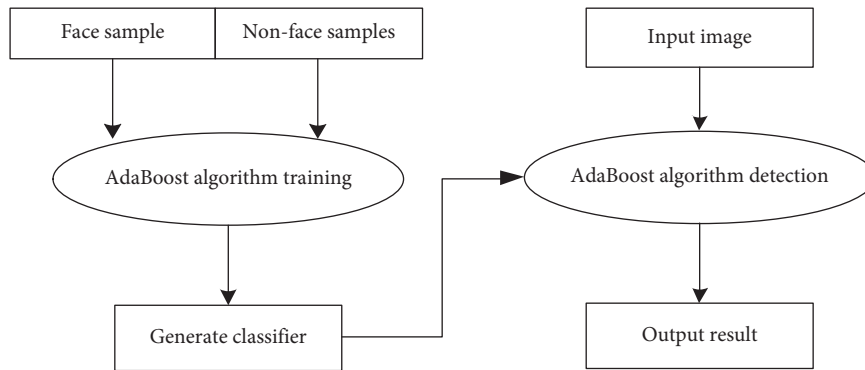
FIGURE 2: Flow chart of facial detection based on AdaBoost algorithm.
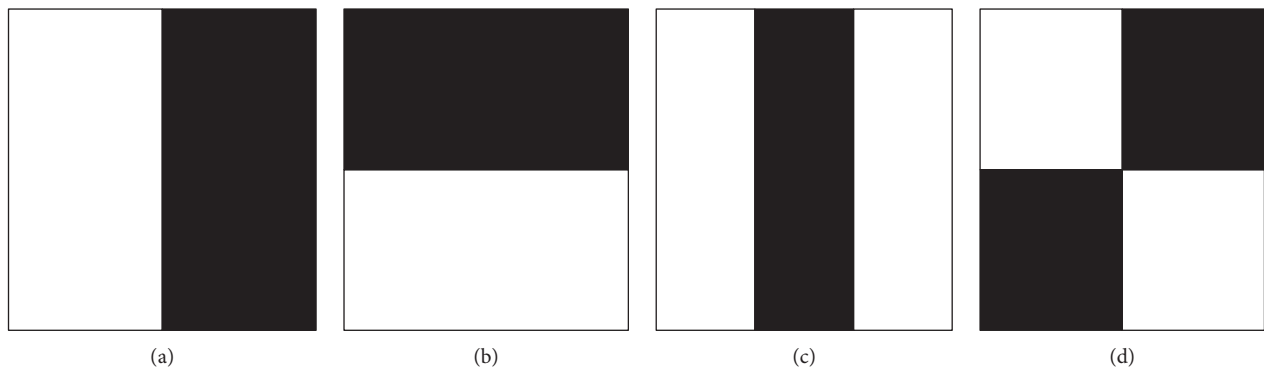

FIGURE 3: Haar-like feature template. (a) Feature template 1, (b) feature template 2, (c) feature template 3, and (d) feature template 4.
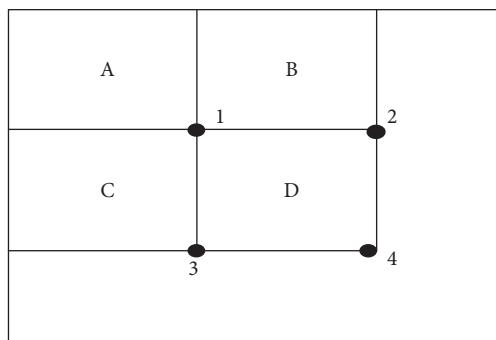

FIGURE 4: Calculation of integral image.

3.2. Expression Feature Extraction Based on Sparse Reserved Projection Algorithm. In the traditional facial expression recognition system, the expression feature extraction module is the core module, and the effective feature extraction directly affects the final recognition result. Facial expressions are rich in features, so there are many methods for feature extraction. This paper mainly uses LBP feature extraction. Local binary pattern (LBP) is a texture feature descriptor commonly used in image processing. Its remarkable feature is that it not only extracts image features effectively but also ensures a small amount of computation. The standard LBP operator contains the gray value of 9 pixels, and its structure is shown in Figure 6.

In Figure 6, g_c is the gray value of the center pixel of the window, which is defined as the threshold value. LBP operator feature extraction can be divided into three steps.

Step 1. Compare the 8 gray values g_0, g_1, ..., g_7 adjacent to the threshold with themselves in turn. If g_i is greater than the threshold, then the position of the pixel corresponding to the gray value is marked as 1; if g_i is less than the threshold value, the position of the corresponding pixel value is marked as 0. Thus, a 3×3 window can generate 8 binaries.

Step 2. Arrange the 8 binary codes generated in Step 1 in sequence to obtain a string of 8-bit binary numbers, namely, the LBP binary code of the pixel in the center of the window.

Step 3. Convert the LBP binary code obtained in Step 2 into decimal number to obtain the LBP value of the pixel in the center of the window.

Although the standard LBP operator can extract the image texture features effectively, its coverage is small and there are some defects. The extended LBP operator and the uniform mode LBP operator evolved from the standard LBP operator, and these two operators have stronger expression ability.

Extended LBP operator is an enhancement of the standard LBP operator, which expands the 3×3 window neighborhood to any scale. The structure of the extended LBP operator is shown in Figure 7. P and RLBP are used to
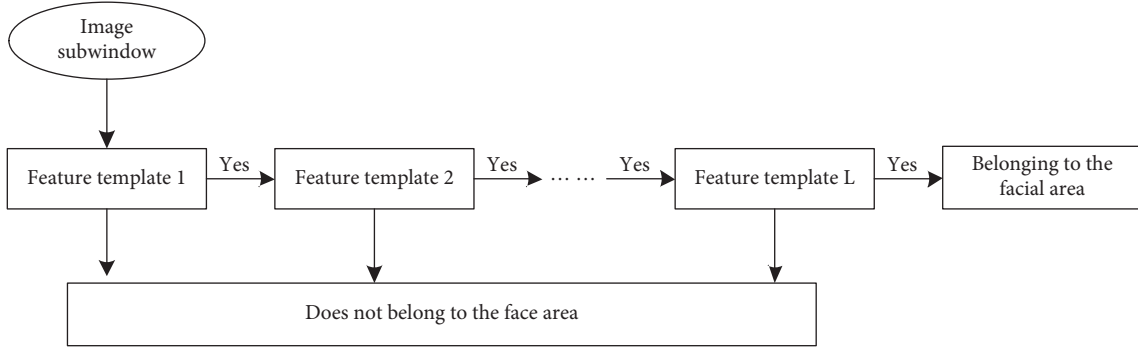
FIGURE 5: Schematic diagram of cascade structure.

represent the LBP value of the center pixel in the circular neighborhood, where P represents the number of sampling points and R represents the radius of the circular neighborhood. Then, the texture information of the three circular neighborhoods in Figure 7 is respectively represented as $LBP_{4,1}$, $LBP_{8,1}$, and $LBP_{8,2}$.

Let the coordinate of the central pixel be (x_c, y_c) and the coordinate of the sampling point in the circular neighborhood be (x_p, y_p), then

$$\begin{cases} x_p = x_c + R \cos\left(\dfrac{2\pi p}{P}\right), \\ \\ y_p = y_c + R \sin\left(\dfrac{2\pi p}{P}\right). \end{cases} \tag{23}$$

In order to describe the texture information of the circular neighborhood conveniently, an auxiliary function $s(x)$ is defined as follows:

$$s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{24}$$

By referring to the calculation method of the standard LBP operator, the LBP value of (x_c, y_c) is

$$LBP_{P,R} = \sum_{P=0}^{P-1} s(g_p - g_c) \times 2^P. \tag{25}$$

In equation (25), $g_p (p = 0, 1, \ldots, P-1)$ represents the gray value of P sampling points around $(x_c, y_c)B$.

4. Experiment Results

In this section, first the parameter settings are discussed followed by results and analysis.

4.1. Parameter Settings. The parameters of the VGG-19 pretraining model were transferred to the facial expression recognition network model by transfer learning method, and the facial expression recognition network model was trained by locality preserving projection (LPP) algorithm. Freeze the parameters of the first 12 convolution layers and train and optimize the last 4 convolution layers, 2 full connection layers, and the new Softmax classification layer. The batch size is set to 48, the initial learning rate is set to 0.005, and the number of training periods epoch is set to 100, respectively.

4.2. Results and Analysis. In order to verify the effectiveness of the method presented in this paper, the expression recognition method based on sparse reserved projection algorithm and the algorithm based on Yang et al. [5] and Zhuang and Ding [6] are respectively used to carry out comparative experiments. From here onward, I refer to them as reference [5] and reference [6] respectively for the ease of understanding.

Figure 8 shows the changes of the loss function values of the two models on CK + expression datasets. It can be seen that, in the process of model learning, the network convergence speed of the algorithm in reference [5] and reference [6] is slow, and the loss function value drops to 0.8 and 0.9, respectively. It shows that the model learning ability of the method in this paper is stronger.

Table 1 shows the confusion matrices of the three models under CK + expression datasets. The recognition rates of the algorithm in reference [5] and reference [6] are 91.17% and 89.25%, respectively, while the recognition rate of the method in this paper is 94.12%. It can be seen that the sparse retained projection algorithm is used in the experiment to achieve a higher recognition rate than other deep learning methods. The proposed method achieves a high recognition rate on the dataset, which shows the rationality and effectiveness of the proposed model.
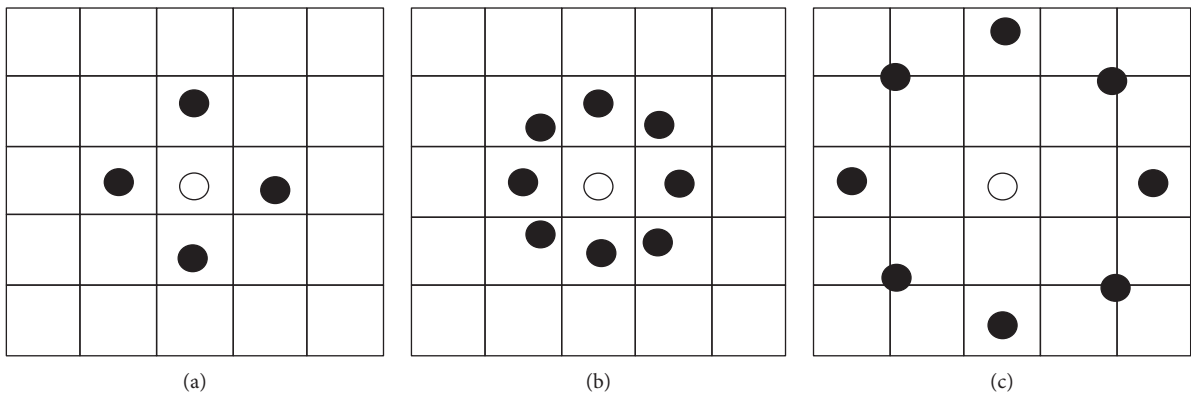
Figure 6: Standard LBP operator.

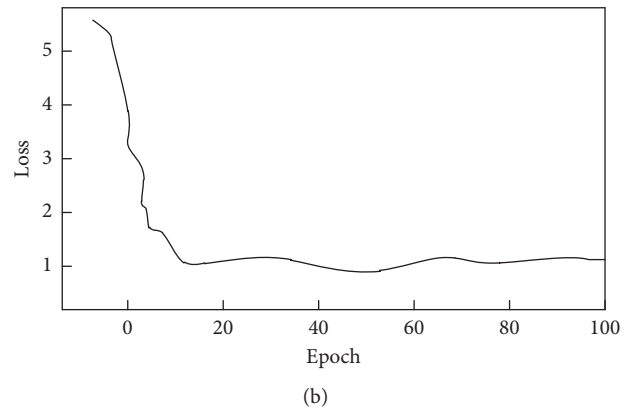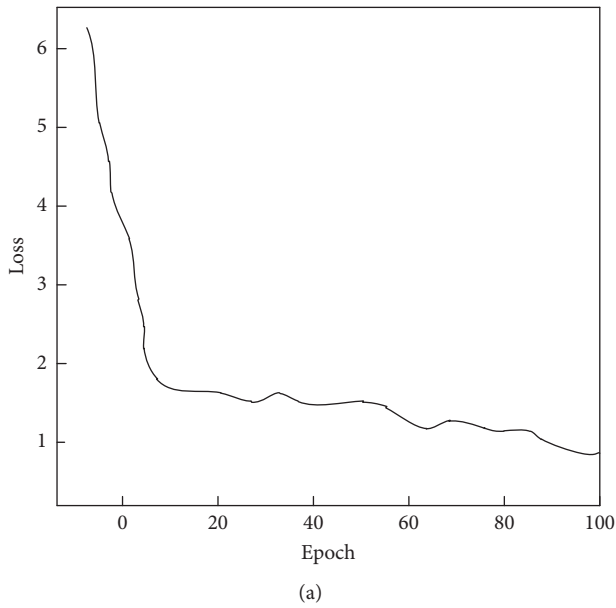Figure 7: Diffusion LBP operator. (a) $LBP_{4,1}$, (b) $LBP_{8,1}$, and (c) $LBP_{8,2}$.
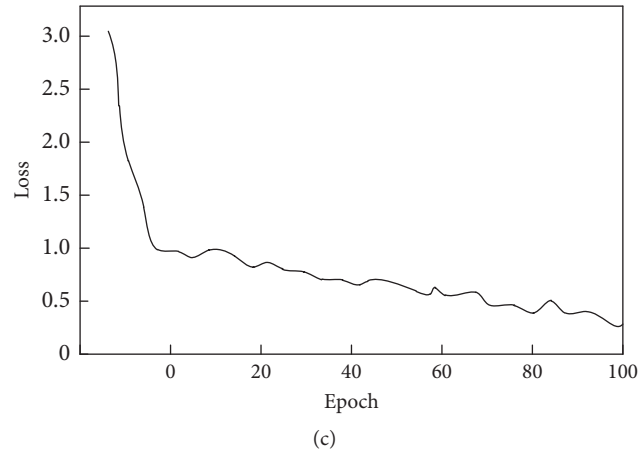

Figure 8: Continued.

Figure 8: CK + loss function value, period number curve. (a) Reference [5] algorithm, (b) reference [6] algorithm, and (c) proposed algorithm.

Table 1: Comparison of recognition rates of different methods.

Algorithm	Recognition rate (%)
Reference [5] algorithm	91.17
Reference [6] algorithm	89.25
Proposed algorithm	94.12

5. Conclusion

In this paper, the locality preserving projection (LPP) algorithm and sparse representation method are studied, and they are combined to carry out face recognition. By using the locality preserving projection (LPP) algorithm to reduce the dimensionality of the high-dimensional face data, this method can maintain the consistency of the local neighbor relationship between samples before and after dimensionality reduction and effectively ensure the invariance of the relationship between the high-dimensional data in the projection to the low-dimensional space. After dimensionality reduction, the sparse representation method is used to classify and match the feature data to achieve recognition effect. Experimental results show that the proposed method achieves a high recognition rate on CK + expression datasets.

Data Availability

The datasets used and/or analyzed during the current study are available from the author on reasonable request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

Acknowledgments

References

[1] Z. Zhang, C. Lai, H. Liu, and Y.-F. Li, "Infrared facial expression recognition via Gaussian-based label distribution learning in the dark illumination environment for human emotion detection," *Neurocomputing*, vol. 409, pp. 341–350, 2020.

[2] M. K. Lee, D. Y. Choi, D. H. Kim, and B. C. Song, "Visual scene-aware hybrid neural network architecture for video-based facial expression recognition," in *Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pp. 1–8, IEEE, Lille, France, May 2019.

[3] Y. Zhu, X. Ni, H. Wang, and Y. Yao, "Face recognition under low illumination based on convolutional neural network," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 13, no. 3, pp. 260–272, 2020.

[4] H. Liu, Y. Chen, W. Zhao, S. Zhang, and Z. Zhang, "Human pose recognition via adaptive distribution encoding for action perception in the self-regulated learning process," *Infrared Physics & Technology*, vol. 114, Article ID 103660, 2021.

[5] Y. Yang, H. Zhang, D. Yuan et al., "Hierarchical extreme learning machine based image denoising network for visual Internet of things," *Applied Soft Computing*, vol. 74, pp. 747–759, 2019.

[6] P. Zhuang and X. Ding, "Divide-and-conquer framework for image restoration and enhancement," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 830–844, 2019.

[7] N. Wang, Y. Wang, and M. J. Er, "Review on deep learning techniques for marine object recognition: architectures and algorithms," *Control Engineering Practice*, vol. 23, Article ID 104458, 2020.

[8] H. F. García, M. A. Álvarez, and Á. A. Orozco, "Dynamic facial landmarking selection for emotion recognition using Gaussian processes," *Journal on Multimodal User Interfaces*, vol. 11, no. 4, pp. 327–340, 2017.

[9] R. Siddiqui, F. Shaikh, P. Sammulal, and A. Lakshmi, *An Improved Method for Face Recognition with Incremental Approach in Illumination Invariant Conditions*, Springer, Singapore, 2021.

[10] J. W. Soh, J. S. Park, and N. I. Cho, "Joint high dynamic range imaging and super-resolution from a single image," *IEEE Access*, vol. 7, pp. 177427–177437, 2019.

[11] P. M. Kumar, B. Poornima, H. Nagendraswamy, M. C, and B. Rangaswamy, "A refined structure preserving image abstraction framework as a pre-processing technique for desire focusing on prominent structure and artistic stylization," *Vietnam Journal of Computer Science*, vol. 10, pp. 1–55, 2021.

[12] S. Madhavan and N. Kumar, "Incremental methods in face recognition: a survey," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 253–303, 2021.

[13] N. Nourbakhsh Kaashki and R. Safabakhsh, "RGB-D face recognition under various conditions via 3D constrained local model," *Journal of Visual Communication and Image Representation*, vol. 52, pp. 66–85, 2018.

[14] D. Li, D. Jiang, R. Bao, L. Chen, and M. K. Kerns, "Crack detection and recognition model of parts based on machine vision," *Journal of Engineering Science & Technology Review*, vol. 12, no. 5, 2019.

[15] Y. Wang, W. Song, G. Fortino, L.-Z. Qi, W. Zhang, and A. Liotta, "An experimental-based review of image enhancement and image restoration methods for underwater imaging," *IEEE Access*, vol. 7, pp. 140233–140251, 2019.