Face Recognition Based on Fusion Feature of LBP and PCA with KNN

Bo Zhai¹ and Zi-mei Li²,*

¹Beijing Information Technology College, Beijing, China
²Changchun Institute of Technology, Changchun, Jilin, China
*Corresponding author

Keywords: Face recognition, Local binary pattern, Principal component analysis, KNN.

Abstract. Face recognition mainly include face feature extraction and recognition (classification) two parts. The basic idea of local binary pattern (LBP) use texture description operator to build a number of local face description, then describe the local combination form global description. LBP operator works with the 3*3 neighborhood pixels for texture description. After scanning the whole image, we get a LBP response images. Then, we use principal component analysis (PCA) method to dimension reduction for response image, finally using K-Nearest Neighbor classifier to classify the test sample. The experimental results show that LBP-PCA-KNN method gains higher recognition rate than PCA-KNN method’s recognition rate, can be effectively used for face recognition.

Introduction

Face recognition mainly include face feature extraction and recognition (classification) two parts. Feature extraction is the mapping process of face data from the original input space to the new feature space, taking the right way to extract face feature, such as size, location, and profile information. Face image data is high-dimensional data, in order to improve the speed of operation, transmission, matching and retrieval, we must implement the original face data dimension reduction while feature extraction, the face data is compressed, with less data to represent the information as much as possible. Currently, the mainly dimension reduction methods are the principal component analysis (PCA), independent component analysis, Fisher linear discriminant method, elastic graph matching method, Gabor wavelet representation, and wavelet dimension reduction method, etc. Face discriminant is according to the result of the feature extraction, comparing test face feature data with training face, calculating the similarity degree to determine the test status. The key of this part is feature selection and classifier design, mainly includes two process training and testing, training part extracts and stores the feature library from the training sample, and testing part extracts the feature from the test sample, then use the trained classifier for test sample. Face discriminant methods mainly include geometrical features method, template matching method, face method, neural network and hidden Markov method and support vector machine (SVM) method, etc. Face image database plays an important influence on face recognition. It is very difficult to develop an algorithm correctly in all possible changes, so almost all of the research is in under certain constraints[1]. On the current, international typical face database are FERET face database, Yale face database, PIE face database, ORL face database, AR face database and so on[2].

Related Works

Local Binary Pattern

Ojala et al. [3] introduced the LBP texture operator in 1996, which originally works with the 3*3 neighborhood. The pixel values of eight neighbors are decided by the value of the center pixel, then, the so-threshold binary values are weighted by powers of two and summed to obtain the LBP code of the center pixel. Fig. 1 shows an example of the LBP operator. In fact, let $g_c$ and $g_0, ..., g_7$ denote respectively the gray values of the center and its eight-neighbor pixels, then the LBP code for the center pixel with coordinate (x, y) is calculated by (1)
LBP(x, y) = \sum_{p=0}^{7} s(g_c - g_p) \tag{1}

where \( s(z) \) is the threshold function

\[ S(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z \leq 0 \end{cases} \tag{2} \]

In traditional real tasks, the statistic representation of LBP codes, LBP histogram (LBPH), usually is used. That is, the LBP codes of all pixels for an input image are collected into a histogram as a texture descriptor, i.e.

\[ \text{LBPH}(i) = \sum \delta\{i, \text{LBP}(x, y)\} \], \( i = 0, \ldots, 2^7 \), \( \tag{3} \)

where \( \delta(\ ) \) is the Kroneck product function.

One extension of the LBP operator is to use neighborhoods of different sizes[4]. The extension is able to take any radius and neighbors around a center pixel, denoted by \( \text{LBP}_{p,R} \), by using a circular neighborhood and the bilinear interpolation whenever the sampling point does not fall in the center of a pixel. For example, \( \text{LBP}_{16,2} \) refers to 16 neighbors in a neighborhood of radius 2. Fig. 2 shows an example with different radii and neighbors. Another extension is the so-called uniform patterns [3], denoted by \( \text{LBP}_{p,R,u2} \). A LBP binary code is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered as a circular. For example, 00000000, 00011110 and 10000011 are uniform patterns. For the computation of LBPH, the uniform patterns are used such that each uniform pattern has an individual bin and all non-uniform patterns are assigned to a separate bin. So, with 8 neighbors, the numbers of bins for standard LBPH are 256 and 59 for uniform patterns LBPH, respectively; with 16 neighbors, the numbers of bins are 65,536 and 243, respectively. Clearly, the uniform patterns are able to reduce the length of histogram vectors.

**Principal Component Analysis**

The essence of PCA is finding the best projection direction which represents the original data in the condition of least mean-square. The dimension of feature space, which decides the size of data, can be reduced by PCA. Assume that face image \( I(x, y) \) is the size of \( M \times N \) gray image, each column are linked together to form a \( D = M \times N \) columns vector. The dimensions of the face image vector is \( D \), the dimensions of the image space is also \( D \). Set the number of training samples is \( n \) and \( x_i \) is the column vector of the \( i_{th} \) face image, all of the training sample covariance matrix can be calculated by the (4).

\[ s = \sum_{i=1}^{N}(X_i - \psi)(X_i - \psi)^T \]

\[ \psi = \frac{1}{N} \sum_{i=1}^{N} X_i \]

\[ \psi \] is the mean image of the training set. Let \( \Lambda = [x_1 - \psi, x_2 - \psi, \ldots, x_n - \psi] \), then \( S = \Lambda \Lambda^T \), the eigenvalues and eigenvectors are then calculated from the covariance matrix. Let \( P = (p_1, p_2, \ldots, p_r) \) (\( r < N \)) be the \( r \) normalized eigenvectors corresponding to \( r \) largest eigenvalues. Each of the \( r \)
eigenvectors is called an eigenface. Now, each face image of the training set is projected into the eigenfaces space to obtain its corresponding eigenface based feature $Y_k$, which is defined as:

$$Y_k = P^T z_k, \quad k = 1, 2, \cdots, N$$

(5)

PCA extracts the global grayscale features of a whole image and the global features are useful and important. But the global feature of face image is environment sensitive.

Considering the face is non-rigid and the compute capacity, we select some local features to assist the global features to resolve this problem. Therefore, local binary pattern (LBP) for local feature extraction is introduced in this article.

**Face Recognition**

The KNN classification approach is quite simple: given a testing sample, the system finds the K-nearest neighbors among training samples, and uses the classes of the K-nearest neighbors to weight class candidates. The similarity score of each nearest neighbor sample to the test samples is used as the weight of the classes of the neighbor sample. If several of K-nearest neighbors share a class, then the per-neighbor weights of that class are added together, and the resulting weighted sum is used as the likelihood score of that class with respect to the testing sample. By sorting the scores of candidate classes, a ranked list is obtained for the test sample. Formally, the decision rule in KNN classification can be written as[6]:

$$core(d, c_i) = \sum_{d_j \in KNN(d)} Sim(d, d_j) \delta(d_j, c_j)$$

(6)

Above, KNN(d) indicates the set of K-nearest neighbors of sample d. $d(d_j, c_i)$ is the classification for sample $d_j$ with respect to class $c_i$, that is,

$$\delta(d_j, c_j) = \begin{cases} 1 & d_j \in c_i \\ 0 & d_j \not\in c_i \end{cases}$$

(7)

For test sample d, it should be assigned the class that has the highest resulting weighted sum.

![Face recognition process](image)

Figure 3. Face recognition process.

After the whole above analysis, the face recognition process is design as in Fig. 3.

**Experimental Result**

In this section, we conducted a series of experiments on standard face library of Yale and AR. Yale face database composed of 165 pieces of face image of 15 people, each with 11 images. The 11 images were taken from different face expression, can be rotate 20 degrees on plane and depth of and face scale has change as much as 10%. The resolution of the image is 100 x100. Fig. 4(a) shows 11 pieces of faces image of the first person in Yale face database. AR face database contains 2600 images, a total of 100 people, men and women each 50 people, the resolution of the image is 165*120. Figure 4(b) shows 13 pieces of the face image of the first person in AR database. $m/P_n$ indicates that m images per individual are randomly selected for training and the remaining n images are used for testing. To compare PCA – KNN with LBP-PCA-KNN, tests for each algorithm are carried out 50 times and the final performance is obtained by averaging the results of all 50 tests.
Experiment Steps
We compared the performance of PCA-KNN with LBP-PCA-KNN on Yale and AR database. Such as, choose randomly 2 pieces of a person's face as a test set in Yale face database, the nine people as the training set. We designed 7 group experiments, the number of training samples is 2,3,4,5,6,7,8 respectively and the number of the corresponding test sample is 9,8,7,6,5,4,3 respectively. Each group runs 50 iterations respectively and its average is taken as the final result.

The main steps of LBP-PCA are as follows:
1) Read the face images data, extract feature using LBP operator on all face gaining LBP feature matrix.
2) Use PCA to train a group of training sample set, gained transformation matrix.
3) Use transformation matrix to test set, and use KNN classifier to face recognition classify.

Fig. 5(a) shows 10 pieces of result images of the first person in Yale database after LBP feature extraction. Fig. 5 (b) shows 13 pieces of result images of the first person in AR database after LBP feature extraction.

The training sample number m is 8 and test sample n is 2 (on Yale), the figure 6(a) shows recognition rate of 50 iterations experiments; The training sample number m is 8 and test sample n is 5(on AR), the figure 5(b) shows line chart of the recognition rate of 50 iterations experiments. As shown, we can see that the LBP-PCA-KNN effect is better than PCA-KNN effect on both database.

Recognition Accuracy Test
First we conduct Recognition accuracy test on Yale face database. Table 1 shows the average recognition rate to compare the performance of PCA-KNN with LBP-PCA-KNN. The maximum
average recognition accuracy and the standard deviation across 50 runs of tests of each algorithm are shown in Table 1. Then, we conduct Recognition accuracy test on the AR database. Table 2 shows the average recognition rate to compare the performance of PCA-KNN with LBP-PCA-KNN. The maximum average recognition accuracy and the standard deviation across 50 runs of tests of each algorithm are shown in Table 2.

Table 1. Recognition accuracy (%) on Yale database (mean).

<table>
<thead>
<tr>
<th>Methods</th>
<th>G_2/P_9</th>
<th>G_3/P_8</th>
<th>G_4/P_7</th>
<th>G_5/P_6</th>
<th>G_6/P_5</th>
<th>G_7/P_4</th>
<th>G_8/P_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA+KNN</td>
<td>65.91</td>
<td>71.52</td>
<td>73.85</td>
<td>74.33</td>
<td>76.19</td>
<td>76.10</td>
<td>75.51</td>
</tr>
<tr>
<td>LBP+PCA+KNN</td>
<td>63.91</td>
<td>76.55</td>
<td>81.71</td>
<td>83.86</td>
<td>86.61</td>
<td>87.87</td>
<td>86.80</td>
</tr>
</tbody>
</table>

Table 2. Recognition accuracy (%) on AR database (mean).

<table>
<thead>
<tr>
<th>Methods</th>
<th>G_2/P_11</th>
<th>G_3/P_10</th>
<th>G_4/P_9</th>
<th>G_5/P_8</th>
<th>G_6/P_7</th>
<th>G_7/P_6</th>
<th>G_8/P_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA+KNN</td>
<td>22.20</td>
<td>25.63</td>
<td>29.39</td>
<td>31.69</td>
<td>33.39</td>
<td>34.71</td>
<td>36.65</td>
</tr>
<tr>
<td>LBP+PCA+KNN</td>
<td>34.32</td>
<td>47.68</td>
<td>56.82</td>
<td>65.34</td>
<td>72.73</td>
<td>77.65</td>
<td>81.35</td>
</tr>
</tbody>
</table>

Figure 7. Recognition accuracy (%) of different algorithms (a) Yale, (b) AR.

Various algorithms’ recognition rates on Yale and AR face database are shown in Figure 7. From the Figure 7, we can see that LBP-PCA-KNN outperforms PCA-KNN with different numbers of training samples per individual.

Conclusions

In this paper, we propose a new method LBP-PCA-KNN to face recognize. The methods of PCA, LBP and KNN are introduced in II respectively. We carried out a series of experiments using our new method LBP-PCA-KNN on Yale and AR database. Through analysis of the experiment data, we conclude that the recognition rate of LBP-PCA-KNN method is higher than PCA-KNN method’s.

Acknowledgement

This research was financially supported by Science and technology project of Beijing Education Committee.

References

