Facial Recognition with PCA and Machine Learning Methods

Jiachen Chen and W. Kenneth Jenkins
School of Electrical Engineering and Computer Science
Pennsylvania State University
University Park, PA
jenkins@engr.psu.edu

***Abstract* - Facial recognition is a challenging problem in image processing and machine learning areas. Since widespread applications of facial recognition make it a valuable research topic, this work tries to develop some new facial recognition systems that have both high recognition accuracy and fast running speed. Efforts are made to design facial recognition systems by combining different algorithms. Comparisons and evaluations of recognition accuracy and running speed show that PCA + SVM achieves the best recognition result, which is over 95% for certain training data and eigenface sizes. Also, PCA + KNN achieves the balance between recognition accuracy and running speed.**

Keywords - Facial Recognition, Principle Component Analysis, Eigenface, Linear Discriminant Analysis, K Nearest Neighbor, Support Vector Machine

I. INTRODUCTION

Facial Recognition is always a hot topic in image processing and machine learning area. In one dimension, it could be widely applied in the areas of biometrics, information security, law enforcement and access control. To increase the system security and reliability, more traditional access control ways, such as keys, PINs or ID cards, are replaced by facial recognition. Thus, the accuracy of facial recognition is essential for these applications. In another dimension, with the development of machine learning in these years, more algorithms could be developed and more calculating resources could be used to decrease the running time, increase recognition accuracy and explore more application scenes.

In the past much of the work of facial recognition focused on detecting the individual features of the face, such as eyes, nose, mouth and head outlines [1]. They tried to define a face model by the position, size and relationship among these features. Such methods are hard to apply for multiple views because with the increasing size of the databases, their low recognition accuracy would decrease and requirements of hardware would increase exponentially. Furthermore, these methods require an ideal start to build the face model, or it would have quite negative results [2].

In this paper, to represent each individual face exactly, a linear combination of eigenfaces from principal component analysis (PCA) is used instead of building face models. The size of eigenfaces will be much smaller than the original image. For some cases this approach could shorten the running time and make the recognition accuracy become higher. In order to evaluate the real recognition accuracy of the newly proposed machine learning model, the databases are divided into training data and testing data. After applying three different methods of classifications, it was possible to compare and evaluate the recognition accuracies and running speed.

II. BACKGROUND AND RELATED WORK

In this paper the "ORL Database of Faces" [3] was used to obtain the image source and then apply tests on it. Principal component analysis (PCA) is used to generate the principal components of an individual face and use them to build the eigenface. After getting the PCA result, three different machine learning classifications were applied: i) linear discriminate analysis (LDA), ii) support vector machine (SVM) and iii) k nearest neighbors (KNN). Then when these two stages of processing has been competed the recognition accuracy and running time of the different methods were produced.

A. ORL Data Base of Faces

The database contains images from 40 individuals and each of them provides 10 different images. The images were taken at different situations, such as the lighting, facial details (glasses/no glasses) and facial expressions (open/closed eyes, smiling/not smiling). Moreover, all the images could tolerant some side movement up to 20 degrees [4]. The size of each image is 92×112 pixels, with 256 grey levels per pixel. Fig. 1 shows three sample images of the same person from the ORL database.

B. Principle Component Analysis (PCA)

In face recognition PCA is used to find the principal components of a given set of images and represent each face image as a smaller size in lower dimensional face space using

Fig. 1. Three sample images of the same person from the ORL database.

the eigenvectors that correspond to higher eigenvalues [5, 6]. The PCA image is much smaller than the initial one but still contains the principal components. In addition, it could increase the recognition speed exponentially.

Assume the initial image I_i has size $m \times n$. The training of the set of N faces can be written as $\mathrm{I} = (I_1, I_2, \dots, I_N)$. The average image A can then be generated as follows,

$$A = \frac{1}{N} \sum_{i=1}^{N} I_i \ . \qquad (1)$$

Next, the vector $Y_i = I_i - A$, is computed, which is the difference image of each face image. Then the covariance matrix C is obtained by using the $Y_i's$ according to

$$\mathrm{C} = \frac{1}{N} \sum_{i=1}^{N} Y_i \, Y_i^T . \qquad (2)$$

Based on (1) and (2), the eigenvectors of the covariance matrix C are then produced by solving the following set of equations,

$$\mathrm{C}V_i = \lambda_i V_i \ . \qquad (3)$$

After finding all the eigenvalues and corresponding eigenvectors, the eigenvectors by eigenvalues are then sorted. The eigenvector associated with the largest eigenvalue shows the largest difference in the image, and the eigenvector associated with the smallest eigenvalue shows the smallest difference. Then the largest K, where K $\leq$ N, can be selected to build the K-eigenvectors $\mathrm{V} = (V_1, V_2, \dots, V_N)$. PCA provides an expression of an input image in terms of a set of basis images that appear as “EigenFaces”. The eigenfaces can then be used to identify the individual face.

C. *Linear Discriminant Analysys (LDA)*

LDA is an optimal linear transformation. It transfers the original data to a much lower dimension space. LDA is used to get a linear transformation that could separate classes in the reduced dimension space.

A data set A can then be denoted as

$$\mathrm{A} = [a_1, a_2, \dots, a_n] = [A_1, A_2, \dots, A_r] \in R^{m \times n}, \qquad (4)$$

where each class I ($1 \leq \mathrm{i} \leq \mathrm{r}$) has $\mathrm{n_i}$ elements, and the total number of data is $\mathrm{n} = \sum_{i=1}^{r} n_i$. $\mathrm{S_B}$ is the between-class covariance matrix given by

$$S_B = \sum_{i=1}^{r} n_i (c_i - c)\,(c_i - c)^T, \qquad (5)$$

and $\mathrm{S_W}$ is the total within-class covariance matrix, given by

$$S_W = \sum_{i=1}^{r} \sum_{j \in N_i} \left(a_j - c_i\right)\left(a_j - c_i\right)^T, \qquad (6)$$

for which $c_i = (\frac{1}{n_i}) \sum_{j \in N_i} a_j$, and $\mathrm{c} = (\frac{1}{\mathrm{n}}) \sum_{j=1}^{n} a_j$ [7].

The optimal dimension reducing transformation $G^T \in R^{l \times m} (l < m)$ for LDA is the one that maximizes the between-class covariance matrix $\mathrm{S_B}$ and minimizes the total within-class covariance matrix $\mathrm{S_W}$ [8].

D. *K Nearest Neighbor (KNN)*

KNN is a well-known method of classification. It is assumed that each class is a cluster and the data points belong to a cluster. Next the locations of clusters need to be determined (cluster center), as well as which data points belong to which cluster. As an unsupervised classification, KNN needs to find the center of each cluster by itself.

Given N data points $\mathrm{x_1}$, $\mathrm{x_2}$, …, $\mathrm{x_N}$, K cluster centers $\mu_1, \mu_2, \dots, \mu_K$ can be found to minimize $\sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \parallel x_n - \mu_j \parallel^2$, where $\mathrm{z_{nk}}$ is 1 if point n belongs to cluster k and is 0, otherwise. This process is then repeated until the label of $\mathrm{z_{nk}}$ can be found that keeps the same as the last round [9, 10]. After selecting the cluster center, the testing data can be applied to get the corresponding class.

E. *Support Vector Machine (SVM)*

SVM was originally proposed by Cortes and Vapnik, and it became increasingly popular after their introduction in the late 1990s, especially as a machine learning classification [11]. SVM is a supervised learning model with associated learning algorithms to analyze and classify data. The core of this method is the selection of the learning algorithm.

Given a set of training examples, each of them is marked as belonging to one class. An SVM training algorithm will build a model to assign a new example to one class. As an SVM model, it is a representation of the examples as points in space and maps these examples to classes, which are divided by a clear gap as wide as possible. New examples can be mapped into the same space and predicted to belong to the class based on which side of the gap they fall [12].

III. METHODOLOGY

The facial recognition system is divided into two stages: i) PCA processing and ii) machine learning classification. The total process is shown in Fig. 2.

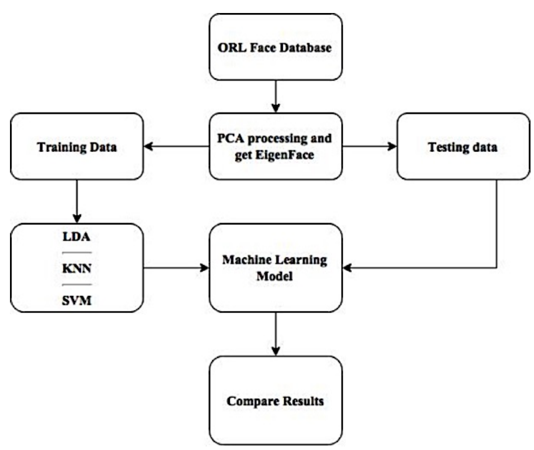

Fig. 2 Illustration of facial recognition two-stage processing.

A. *Divide the Data Base*

The size of the ORL faces databases are 40×10, 40 different people for which each has 10 different images. The size of training and testing data will affect the detecting accuracy. Larger training data and smaller testing data should have better performance. However, in real application, the size of training data will always be smaller than the testing data. In order to determine the effect of the size of training data, the data are divided into three different sizes of training groups and testing groups. The training group size could be 3, 5, 7, and the corresponding testing group size is 7, 5, 3.

B. *PCA Processing*

To reduce the calculating time and obtain the eigenfaces, the PCA is applied on the original image that has a size of 92 × 112 pixels. Different sizes of the eigenface can then be obtained, such as 5×5, 20×20 or 80×80. Fig. 3 shows the original image and the corresponding eigenface. The size of the eigenface is 20×20. The larger size of eigenfaces will require more running time. In this paper, different sizes of output eigenfaces were used to find the relationship between the size of eigenface and the recognition accuracy.

Fig. 3 Original image and eigenface.

IV. EXPERIMENTAL RESULTS

In the following experiments results are compared and evaluated in three dimensions as follows:

(1) Keep the size of eigenface and change the size of training data, and then determine the effects of size of training data on the recognition accuracy.
(2) Keep the size of training data and change the size of eigenface, and then discover how the size of eigenface influences the recognition accuracy.
(3) Compare the running time of different classifications, from different sizes of training data and different sizes of eigenface. Since the running speed is an equally important factor as recognition accuracy in real applications, attempts are made to find a method that balances the running speed and recognition accuracy.

A. *The Influence of the Size of Eigenface*

The next experimental steps were to keep the size of eigenface to be 5*5, divide the training data size to 3, 5, 7 and the corresponding test data size is 7, 5, 3. After repeating the process 10 times, the mean and standard deviation (SD) were obtained from LDA, KNN and SVM. Table 1 shows the results.

From Table 1, it was discovered that a larger size of training data creates a higher recognition accuracy. Also, from the results in Table 1, it can be predicted that as the training data gets larger more precise high eigenvalues and valuable eigenfaces could be found. Although the LDA has the relative lower recognition accuracy, it keeps the low level of standard deviation, which means that LDA has more predictable results. KNN has the average level of recognition accuracy and the highest standard deviation. It shows that the result of KNN is not stable. SVM has the highest recognition accuracy and

Table 1 Recognition result from different size of training data

		LDA		KNN		SVM	
		mean	SD	mean	SD	mean	SD
Training data size	3	63.82%	1.54%	69.64%	4.51%	71.04%	3.04%
	5	66.50%	2.04%	77.60%	2.09%	80.25%	2.06%
	7	95.33%	2.19%	93.58%	2.15%	95.75%	1.90%

acceptable standard deviation. From this dimension, SVM is the most suitable method to apply for facial recognition.

B. *The Influence of the Size of the Training Data*

Next the size of training data was kept constant the size of eigenfaces was changed. Fig. 4 shows the different eigenfaces that were obtained from the same individual. Fig. 4(a) is 5×5, (b) is 20×20, and (c) is 80×80. We could see that larger size of eigenfaces is more vague, which means that when more eigenvectors are used fewer details remain.

(a) 5×5 (b) 20×20 (c) 80×80

Fig. 4 Eigenface of different size

After repeating the process for 10 times, the mean and standard deviation (SD) were again obtained from LDA, KNN and SVM. Table 2 shows the results.

Table 2 Recognition result from different size of eigenface

		LDA		KNN		SVM	
		mean	SD	mean	SD	mean	SD
Eigen-face Size	5×5	66.50%	2.04%	77.60%	2.09%	80.25%	2.06%
	20×20	92.75%	1.11%	90.40%	2.39%	93.50%	2.57%
	80×80	94.60%	1.29%	85.55%	1.82%	93.65%	1.81%

From Table 2 it can be found that a larger size of eigenface will not absolutely bring higher recognition accuracy. With LDA, the recognition accuracy reaches the highest of all of three methods while the size of eigenfaces is 80×80. However, the recognition accuracy is the lowest at the same time while the size of eigenfaces is 5×5. The standard deviation keeps low, and it's always the strength of LDA. KNN has strange results, the 80×80 eigenface does not bring higher recognition accuracy than 20×20 one, and the overall recognition accuracy is the lowest of three methods. SVM almost has the highest recognition accuracy of different sizes of eigenfaces, and the standard deviation keeps acceptable, higher than LDA, but almost equals to KNN.

Thus, based on the analysis above, SVM is the best method to apply for facial recognition, and 20×20 eigenface is a reasonable size, as the recognition accuracy does not increase obviously in 80×80 ones.

C. Running Time Comparison

Furthermore, considering the real application the running speed is also an important factor to consider when selecting the recognition algorithms. Different running times of LDA, KNN and SVM are obtained from different sizes of eigenface and different sizes of training data. To keep the precision of the results, the process is repeated 10 times to produce the average time. The results are shown in Table 3.

Table 3 Average Running Time

Average running time(s)		LDA			KNN			SVM		
Training Group Size		3	5	7	3	5	7	3	5	7
Eigen-Face Size	5×5	1.14	0.97	1.07	0.92	0.89	0.84	3.59	3.51	3.42
	20×20	1.08	1.01	1.10	0.76	0.85	0.86	4.07	4.07	4.02
	80×80	1.20	1.22	1.17	0.83	0.91	0.79	4.59	4.97	5.03

From Table 3 it can be found that, when the size of eigenface becomes larger, the average running times of most cases of the three methods are increasing. Except if the size of eigenface is 20×20, the average running speed may be faster than both larger and smaller eigenface. However, the average running time does not follow the proportion of the size of training data. When comparing with the size of training data and eigenface, it can be seen that when the size of training data is 5 and the size of eigenface is 20×20, the fastest running speeds result.

From Table 3 it can be determined that when the size of eigenface becomes larger the average running times of most of the three methods are increasing. Except if the size of eigenface is 20×20, the average running speed may be faster than both larger and smaller eigenface. However, the average running time does not follow the proportion of the size of training data. Comparing with the size of training data and eigenface, it is apparent that when the size of training data is 5 and the size of eigenface is 20×20, again the fastest running speeds are obtained.

Among these classifications, KNN has the fastest running speed and SVM has the lowest running speed. LDA is a little slower than KNN but much faster than SVM. Thus, although SVM always has higher recognition accuracy the running speed is much slower than other two classifications.

V. Summary and Conclusions

This paper presents three different methods of facial recognition. Based on the comparison above that keeps the size of training data to be 5, which is half size of the total data, and obtaining the eigenface with a size off 20×20, an acceptable recognition accuracy and fastest running speed can be obtained. If there is little sensitivity to running speed and a strong desire to get higher recognition accuracy, it appears that SVM is the best choice. Otherwise, KNN could achieve the better balance between running speed and recognition accuracy.

In future studies, efforts could be made to apply other machine learning classifications and combine some of them to build more complex system. They should have higher recognition accuracy, and they could deal with larger sizes of data.

VI. References

[1] Turk, Matthew, and Alex Pentland, "Eigenfaces for recognition." *Journal of cognitive neuroscience* 3.1 (1991): 71-86.

[2] Parmar, Divyarajsinh N., and Brijesh B. Mehta, "Face recognition methods & applications." arXiv preprint arXiv: 1403.0485 (2014).

[3] "The ORL Database of Faces": http://www.cl.cam.ac.uk/ research/dtg/attarchive/facedatabase.html

[4] Dandpat, Swarup Kumar, and Sukadev Meher, "Performance improvement for face recognition using PCA and two-dimensional PCA." *Computer Communication and Informatics (ICCCI), 2013 International Conference on*. IEEE, 2013.

[5] Jindal, Navneet, and Vikas Kumar, "Enhanced face recognition algorithm using pca with artificial neural networks." *International Journal of Advanced Research in Computer Science and Software Engineering* 3.6 (2013).

[6] Yang, Jian, et al, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition." *IEEE transactions on pattern analysis and machine intelligence* 26.1 (2004): 131-137.

[7] Anzai and Yuichiro, *Pattern recognition and machine learning*. Elsevier, 2012.

[8] Park, Cheong Hee, and Haesun Park. "A comparison of generalized linear discriminant analysis algorithms," *Pattern Recognition* 41.3 (2008): 1083-1097.

[9] Liu, Zhun-Ga, Quan Pan, and Jean Dezert. "A new belief-based K-nearest neighbor classification method," *Pattern Recognition* 46.3 (2013): 834-844.

[10] Wu, Yingquan, Krassimir Ianakiev, and Venu Govindaraju. "Improved k-nearest neighbor classi-fication," *Pattern recognition* 35.10 (2002): 2311-2318.

[11] Brereton, Richard G., and Gavin R. Lloyd. "Support vector machines for classification and regression," *Analyst* 135.2 (2010): 230-267.

[12] Abe and Shigeo, *Support vector machines for pattern classification*, Vol. 2. London: Springer, 2005.