

Research Article

Research on Face Recognition Based on Embedded System

Hong Zhao, Xi-Jun Liang, and Peng Yang

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Correspondence should be addressed to Hong Zhao; 594286500@qq.com

Received 8 July 2013; Revised 5 September 2013; Accepted 25 September 2013

Academic Editor: Wuhong Wang

Copyright © 2013 Hong Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because a number of image feature data to store, complex calculation to execute during the face recognition, therefore the face recognition process was realized only by PCs with high performance. In this paper, the OpenCV facial Haar-like features were used to identify face region; the Principal Component Analysis (PCA) was employed in quick extraction of face features and the Euclidean Distance was also adopted in face recognition; as thus, data amount and computational complexity would be reduced effectively in face recognition, and the face recognition could be carried out on embedded platform. Finally, based on Tiny6410 embedded platform, a set of embedded face recognition systems was constructed. The test results showed that the system has stable operation and high recognition rate can be used in portable and mobile identification and authentication.

1. Introduction

Face recognition technology [1] emerged in 1980s, developed rapidly, and obtained staged achievements since 1990s. Gradually, it has been applied in feature search system, authentication system [2], and access control system [3]. Due to a number of images feature data, complex calculation, and the larger storage space and high processing capacity, currently, most face recognitions are realized only by PCs with high performance; so portability and mobility in this process are restricted greatly. At present, the embedded system [4] is widely used in the front-end of entrance guard system and attendance system in order to collect face images. Then, the information collected is transferred to the back-end over the network, and the face recognition is carried out by the back-end PCs. However, this working mode heavily relies on recognition by the back-end and is limited by the bandwidth and stability of data transmission network. It still cannot reach the purpose of the move at will.

Since portability and mobility in this process are restricted greatly in the current face recognition system, it is necessary to develop a set of face recognition systems in which both image collection and recognition are realized on the embedded system.

2. Principle of Face Recognition

The process of face recognition is divided into two stages, training and recognition stages, shown in Figure 1.

2.1. Face Training. To ensure convenient face image processing, the original YUV format image is transformed to IplImage format image. Haar-like face detection algorithm (Viola-Jones method) is used to identify face region [5]. In order to enhance the contrast of image, reduce the influence from external factors and improve the following recognition rate; the face image identified is processed with the histogram equalization.

In order to obtain main features of original image, Principal Component Analysis [6] (PCA) subspace of eigenfaces, the PCA is used to extract subspace of eigenface from face image processed. This method can effectively reduce redundant data, and data can be processed in a low-dimensional feature space. Meanwhile, most information of the original image is saved.

2.2. Face Recognition. Similarly, the test of face image is processed through format transformation, Haar-like face detection, and histogram equalization.

The face image processed is projected to the PCA subspace of eigenfaces; thus, projection coefficients on the

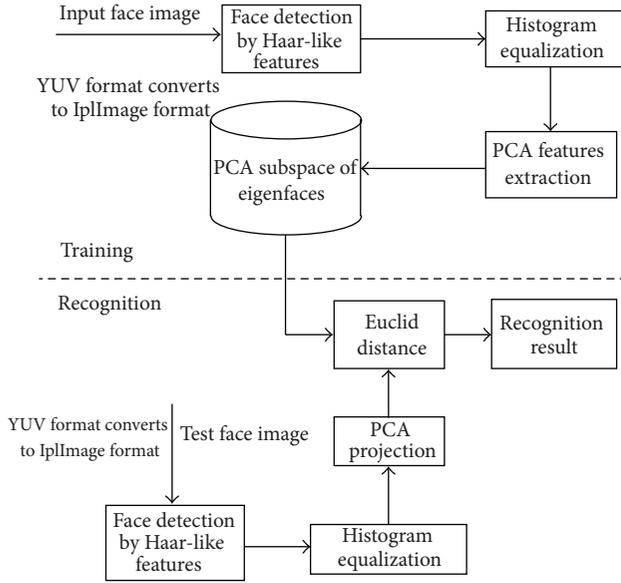


FIGURE 1: Principle of face recognition.

subspace can be obtained. The projection coefficients represent the position of test face image in the PCA subspace of eigenfaces. Then, the analysis contrasts the coefficients with the coefficients of PCA subspace of eigenfaces; finally, the face can be recognized by using Euclidean Distance [7].

3. Major Algorithms

3.1. Haar-Like Face Detection Algorithm

3.1.1. Haar-Like Features. Viola-Jones in 2001 published a paper [8] which was a watershed in the real-time face detection technology. The real-time face detection was realized through combining Adaboost algorithm and Cascade algorithm. Papageorgiou and Viola put forward the original Haar-like features when they applied wavelet transformation to extract features from images. The feature library contained features of three types and four kinds. The three types, two-rectangle feature, three-rectangle feature, and four-rectangle feature, are presented in Figure 2. Since this feature library can only describe the structure with specific directions (horizontal, vertical, and diagonal), the features extracted are relatively rough. Subsequently, Lienhart and Maydt put forward a series of extended Haar-like features [9] as listed in Table 1 based on the basis mentioned above; the edge-feature is extended to 4 types, and linear-feature is extended to 8 types adding 2 center-features. These extended Haar-like features make face recognition more convenient and fast.

Haar feature stands for the differential value of gray level sum between corresponding region of the black rectangle and the white rectangle. And Haar feature reflects the degree of image part-graying. Each feature is composed of 2~3 rectangles, and they are applied in the detection of edge-feature, linear-feature, and center-feature. Value of each

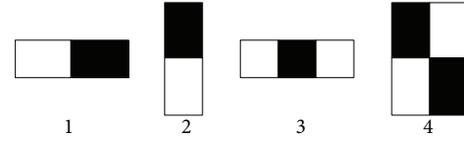


FIGURE 2: Original forms of Haar-like features. (1, 2—edge-feature, 3—linear-feature, 4—diagonal-feature).

feature is made up by the sum of pixel values in corresponding rectangle region, shown as

$$\text{feature}_j = \sum_1^N \omega_i \cdot \text{rectSum}(r_i), \quad (1)$$

where $\text{rectSum}(r_i)$ represents the gray-level integration of image enclosed by rectangular r_i ; N stands for the number of matrices which compose feature j ; ω_i represents the weight of rectangular region; value +1 represents the weight of white rectangular region; value -1 represents the weight of black rectangular region.

3.1.2. Integral Image. Setting the four basic features as an example, permutation, and combination at random in a window with size 24×24 will generate at least hundreds of thousands of features. Calculating eigenvalues of these features will make a large amount of calculation. Therefore, obtaining gray value of the pixel and then making evaluation obviously cannot meet real-time needs. The integral-image method can make fast to process image on calculating the eigenvalues of current subimage.

Integral image is a method of fast calculation of $\text{rectSum}(r_i)$ with the idea of replacing time with space. The sum of pixel values in rectangle region formed from starting point of image to the rest is saved as an array. We can directly use the array to do calculation when we need to calculate $\text{rectSum}(r_i)$ of certain region. This method avoids recalculation of pixels of this region; so calculation speed improves.

3.1.3. Adaboost Algorithm. Adaboost algorithm is a kind of classifier algorithms. Its basic idea is constructing an accurate classifier with strong ability of classification by means of combining a large number of simple classifier according to some rules.

(1) *Training of Simple Classifiers.* The form of simple classifier generated by the j th feature is shown as

$$h_j(x) = \begin{cases} 1, & p_j \cdot f_j(x) < p_j \cdot \theta_j \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where h_j is the value of simple classifier; x is a testing sub-window; θ_j is the threshold; p_j is the attribution of sample and it indicates the direction of the sign of inequality; value +1 represents positive samples and -1 negative samples; and $f_j(x)$ is eigenvalue.

Training samples include positive samples and negative samples. The object samples (human face images) to be

TABLE 1: Three kinds of Haar-like features.

Edge-feature				
Linear-feature				
Center-feature				

detected represent positive samples, and any other images represent negative images. All sample images are normalized to the same size with 20*20.

From formula (2), a weak classifier is decided by both corresponding threshold θ_j and feature f_j . Each feature f_j will be trained to obtain a specified classifier, and the analysis will find the optimal threshold θ_j and minimize the classification error of using this weak classifier to classify all samples under current weight distribution.

For each feature, the corresponding weak classifier is trained. Finally, the weak classifier with lowest classification error ratio for all training samples is selected, called optimal weak classifier.

(2) *Training of Strong Classifiers.* Adaboost classifier includes many optimal weak classifiers which are connected together by some rules and weights. After T times training, T optimal weak classifiers are generated.

A strong classifier has been constructed by T optimal weak classifiers in terms of the following:

$$C(x) = \begin{cases} 1, & \sum_{t=1}^T a_t \cdot h_t(x) \geq \frac{1}{2} \sum_{t=1}^T a_t \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where T represents the number of optimal weak classifiers included in the strong classifier and $a_t = \log(1 - \epsilon_t)/\epsilon_t$, ϵ_t represents error ratio of the t th optimal weak classifiers.

From formula (3), all weak classifiers have their judgments for testing image. This process is similar to "voting." Then through getting the weighted sum of "voting" in terms of error rates of weak classifiers, the final result can be made by comparing the result of weighted sum of "voting" with the result of average "voting." The result of average "voting" is, namely, probability average value under the condition where the supporting probability is equal to objecting probability.

3.1.4. *Cascade Classifier.* Since the detection process of the strong classifier composed of several weak classifiers costs lots of time, Paul Viola and Michael Jones put forward the cascade

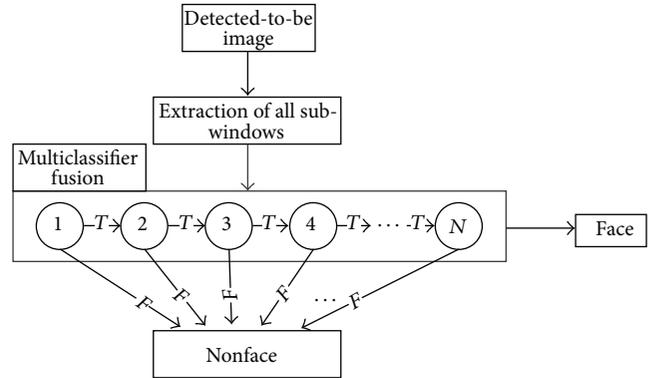


FIGURE 3: Structure of cascade classifier.

face classifier based on Adaboost algorithm. This classifier can detect face quickly and recognize face effectively.

In fact, the multilayer structure proposed by Paul Viola and Michael Jones is a degenerated decision tree. If the current subwindow is a human face when the images go through all simple classifiers in a certain order, the current image will go to the next detection, otherwise, the current sub-window is ceased and the next sub-window will work, as shown in Figure 3.

3.2. *"Eigenface" Recognition Algorithm.* Based on principal component method, "Eigenface" recognition algorithm [10] has been widely applied to face detection and face recognition.

"Eigenface" is the assembly of these eigenvectors corresponding to the large eigenvalues in face covariance matrix. It treats face image as a vector and gets eigenvectors by Karhunen-Loeve transform [11]. The eigenvectors which are similar to the face are called eigenface. The linear combinations of these eigenvectors are used to describe, represent, and recognize the face image.

The preprocessed face image is projected to the subspace composed of "Eigenface". Then projection coefficients on the subspace can be obtained. The projection coefficients that

represent the position of test face image in the PCA subspace of eigenfaces compare with coefficients of the subspace of eigenfaces and finally recognize by using Euclidean Distance.

The method of computing credibility is the key to recognition. Now the best method of computing confidence is base on Euclid Distance, as shown in the following:

$$C = 1.0 - \frac{\sqrt{D / (\text{trainFacesNum} \cdot \text{eigenVectorsNum})}}{255.0}, \quad (4)$$

where C represents credibility; D is the Euclidean Distance between the projection of a test image and the projection of the trained images; trainFaceNum is the number of faces in training; and eigenVectorsNum is the number of face eigenvectors.

4. System Construction

4.1. General Architecture. Based on Tiny6410 embedded platform, a set of embedded face recognition system is constructed. The man-machine interface of the system is programmed with Qt graphic library; the part of video gathering is implemented by video interface of v4l2 [12] in Linux; and OpenCV library is applied in the video processing part.

Two steps in the system work as follows:

- (1) detection stage: system searching for the face region (displayed by rectangle) in the whole image;
- (2) recognition stage: contrasting the face image obtained above to the face image trained in the database, and then judging the person who it is.

If the system recognition is successful, the recognition result will be displayed in white text and the system will pop a dialog box which shows login. If failed, system will pop a warning dialog.

4.2. Working Process

4.2.1. System Training. As shown in Figure 4, test ID is inputted firstly, a frame of image from USB camera is gotten by system, and then the image is grayed and processed with histogram equalization in order to enhance the degree of contrast of image. Next, the preprocessed image is determined whether it will be added to the training set. PCA algorithm is applied to deal with all images in the training set when the number of images in training set reach the presented number. Finally, the XML database is generated.

4.2.2. System Recognition. In the recognition stage, system reads the database file of trained images and applies PCA algorithm to compare test image with the database data. If credibility goes beyond the threshold value, the corresponding user name is displayed on the screen and a message will be popped; otherwise, a warning dialog will be popped.

System can capture several images from camera and compute average credibility of these images in order to improve the accuracy and the reliability.

Figures 5 and 6 are recognition result of pretraining and recognition result of posttraining, respectively.

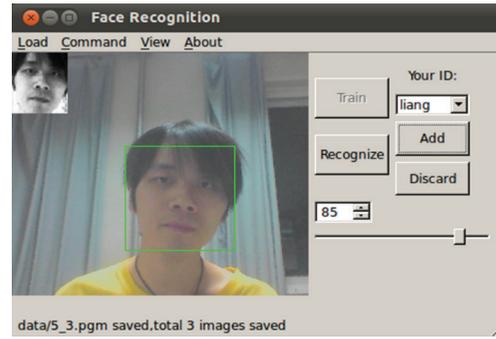


FIGURE 4: Face training.

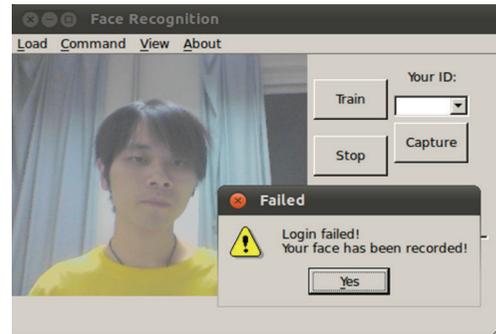


FIGURE 5: Recognition result of pretraining.

5. System Test

The system carries out 9 times tests.

In the No. 1~No. 4 tests, only the face data of person A is added to the face database. At this time, person B and person C are “strangers.” Only Person A can login the system.

In the No. 5~No. 9 tests, the face data of person B is added to the face database. At this time, only person C is a “stranger,” but both person A and person B can login the system.

The results of test are shown as Table 2.

In the 1st test, the time consuming for login person A is 14 s. At this time, system temporarily saves the login information of person A. In the 2nd test, person A can login immediately. In the 3rd and 4th test, since person B and person C did not login before, and person B and person C are not trained, thus B and C cannot login. At this time, system still saves the login information of person A. So person A can login immediately. In the 5th and 6th tests, person B is trained. At this time, person B can login immediately, and system temporarily saves the login information of person B. In the 7th test, because person C did not be trained, person C cannot login. In the 8th test, system still saves the login information of person B when person A login again. So system needs to rescan all sub-windows (time consuming is 12 s), and temporarily saves the login information of person A. In the 9th test, system still saves the login information of person A when person B login again. So system needs to rescan all sub-windows (time consuming is 11 s), and temporarily saves the login information of person B.

TABLE 2: Test results of embedded face recognition system.

No. of test	No. of person	Time for login (s)	No. of frames for recognition	Average credibility	No. of error recognition
1	A	14	2	0.856	0
2	A	—	10	0.866	0
3	B	∞	10	0.675	0
4	C	∞	10	0.778	0
5	A	—	10	0.878	0
6	B	—	10	0.906	0
7	C	∞	10	0.824	0
8	A	12	2	0.883	0
9	B	11	2	0.913	0

(1) Time consuming for login: referring to the time consumed from test user sitting in front of camera to test user login successfully.

(2) Number of frames recognized: referring to the number of frames used to calculate the average credibility.

(3) —: referring to the immediate login.

(4) ∞: referring to the failure of login.

(5) The threshold value of credibility is set as 0.85.

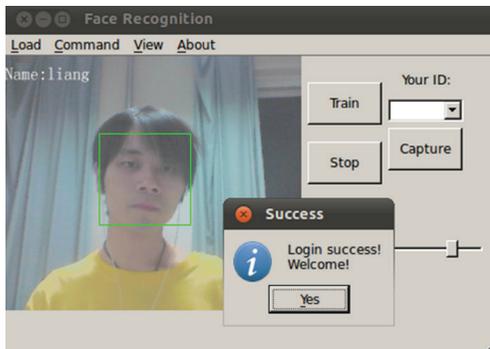


FIGURE 6: Recognition result of posttraining.

6. Conclusion

This paper introduces the specific face recognition technology which is based on embedded platform and puts forward a solution, which stresses on face detection algorithm, face recognition algorithm, and application development. This technology makes full use of the advantage of PCA algorithm on feature extraction and the advantages (such as fast detection speed and high detection rate) of AdaBoost algorithm based on Haar. A set of embedded face recognition system based on Tiny6410 embedded platform is realized. After face recognition testing, the results showed that this system runs stably and has high recognition rate. Thus, it can be widely used in the Things of Internet that needs to verify user identification through portable and mobile methods [13] and in Intelligent Transportation System [9, 10] that needs face recognition technology. In the future research, the Cortex A8 embedded platform that has better ability of floating-point operation will be applied in the system in order to further improve the overall performance of the system.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant no. 61262016, University

Foundation of Gansu Province under Grant no. 14-0220, Natural Science Foundation of Gansu under Grant no. 1208RJZA239, and the Technology Project of Lanzhou (2012-2-64).

References

- [1] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.
- [2] X. Liu, T. Chen, and B. V. K. Vijaya Kumar, "Face authentication for multiple subjects using eigenflow," *Pattern Recognition*, vol. 36, no. 2, pp. 313–328, 2003.
- [3] D. Bryliuk and V. Starovoitov, "Access control by face recognition using neural networks and negative examples," in *Proceedings of the 2nd International Conference on Artificial Intelligence*, pp. 428–436, Crimea, Ukraine, September 2002.
- [4] W. Ping, "Research on the embedded system teaching," in *Proceedings of the International Workshop on Education Technology and Training and the International Workshop on Geoscience and Remote Sensing*, vol. 1, pp. 19–21, Shanghai, China, December 2008.
- [5] M.-T. Pham and T.-J. Cham, "Fast training and selection of Haar features using statistics in boosting-based face detection," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–7, Rio de Janeiro, Brazil, October 2007.
- [6] S. Nedeveschi, I. R. Peter, and A. Mandrut, "PCA type algorithm applied in face recognition," in *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing (ICCP '12)*, pp. 167–171, Cluj-Napoca, Romania, 2012.
- [7] J. Javed, H. Yasin, and S. F. Ali, "Human movement recognition using euclidean distance: a tricky approach," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, vol. 1, pp. 317–321, Yantai, China, October 2010.
- [8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, Kauai, Hawaii, USA, December 2001.
- [9] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International*

Conference on Image Processing (ICIP '02), vol. 1, pp. 900–903, September 2002.

- [10] E. Gumus, N. Kilic, A. Sertbas et al., “Eigenfaces and support vector machine approaches for hybrid face recognition,” *The Online Journal on Electronics and Electrical Engineering*, vol. 2, no. 4, pp. 308–310, 2010.
- [11] M. Effros, H. Feng, and K. Zeger, “Suboptimality of the Karhunen-Loève transform for transform coding,” *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1605–1619, 2004.
- [12] L. Yinli, Y. Hongli, and Z. Pengpeng, “The implementation of embedded image acquisition based on V4L2,” in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC '11)*, pp. 549–552, Ningbo, China, September 2011.
- [13] Z. Hong, S. Chao, and C. Jie, “Integration schemes for IoT application systems with diverse domain,” *Journal of Beijing Institute of Technology*, vol. 32, no. 12, pp. 201–204, 2012 (Chinese).



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

