Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 1 (2012) 276 – 281

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Computational Intelligence and Bioinformatics

LBP Based Fast Face Recognition System on Symbian Platform

Helei Sun[a,*], Jie Shen[a], Bin Chen[a]

[a]*School of Computer Science and Engineering*
University of Electronic Science and Technology of China ,Chengdu, China

Abstract

In this paper, an implementation of a LBP (local binary pattern) based fast face recognition system on symbian platform is presented. First, face in picture taken from camera is detected using AdaBoost algorithm. Second, the pre-processing of the face is done, including eye location, geometric normalization, illumination normalization. During the face preprocessing, a rapid eye location method named ER (Eyeball Search) is proposed and implemented. Last, the improved LBP is adopted for recognition. Although the computational capability of the symbian platform is limited, the experimental results show good performance for recognition rate and time.

Keywords: Face recognition; Symbian platform; Rapid eye location; Eyeball Search; hierarchical multiscale LBP.

1. Introduction

Face recognition has been widely used in different fields. Many face recognition algorithms have gained encouraging performance. Recently, with the popularity of digital cameras in mobile devices increasing rapidly, an increasing attention is given to a new application field of face recognition on mobile phones [1].

This paper introduces the implementation of a LBP based fast face recognition system on symbian platform in detail. The architecture of our system is shown in Fig. 1 (a). The system contains face detection, face pre-processing and face recognition, which are introduced in section 2, section 3 and section 4 respectively.

*Helei Sun. *E-mail address:* shl504@126.com.

doi:10.1016/j.aasri.2012.06.042

2. Face Detection

Face detection is the first step in face recognition system. Its reliability has a major influence on the performance and usability of the entire face recognition system [3]. Reference [4][5] summarized a lot of face detection methods. In this paper, AdaBoost based method is adopted for face detection because so far it is the most successful ones in terms of detection accuracy and speed. This method can detect faces efficiently under illumination and background variation conditions. Fig. 1 (b) shows a result of face detected by AdaBoost.

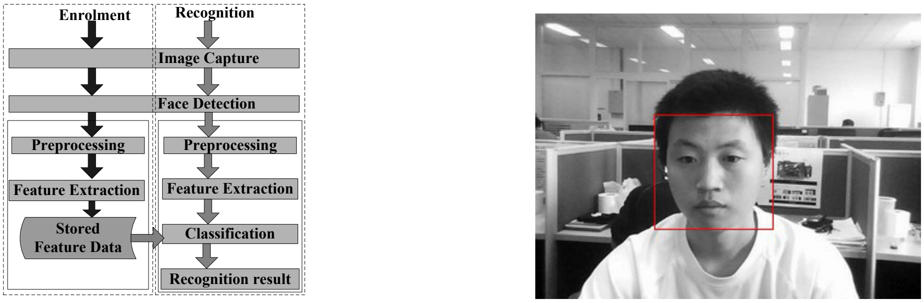


Fig.1. (a) The architecture of our system; (b) A result of face detected by AdaBoost.

3. Face Pre-processing

Face pre-processing is an important step of the face recognition system. Its reliability has an important influence on the recognition rate. Fig. 2 shows some examples of face image detected by AdaBoost. These images differ from each other in background, size, illumination, and so on. Therefore, each image has to be normalized by same standard. The face pre-processing mainly contains geometric normalization, illumination normalization and noise filtering. Generally, the first step of preprocessing is eye location.

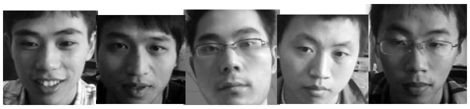


Fig.2. Face images detected by AdaBoost.

3.1. Rapid eye location

There are many methods of eye location. Considering the limited amount of memory and a less powerful processor of mobile phones, some complex and time consuming methods are not suitable. In this paper, a quick and efficient eye location method named ER (Eyeball Search) is proposed and implemented, which is a good trade-off between time and accuracy.

As shown in Fig. 2, these images are detected by AdaBoost. As a face has been detected, we can estimate the area which contains the positions of the two eyes. Then, precise eye location is done by searching eyeballs in the area. The concrete steps of Eyeball Search are given as follows.

1) *Determine the boundary of search area*

If a face image with size $w \times h$ is divided into four regions averagely as shown in the first picture in Fig. 3, in almost all cases, the eyes exist in the first and second region. The center point C with coordinate ($w/2, h/2$) is one boundary point. Through a great deal of experiments, we determine the other two boundary points A and B with the coordinate of (*3w/14, 13h/40*) and (*11w/14, 13h/40*) respectively. So the search area is the area containing the two rectangles demarcated by point A, B and C. See Fig. 3 (a) for details.

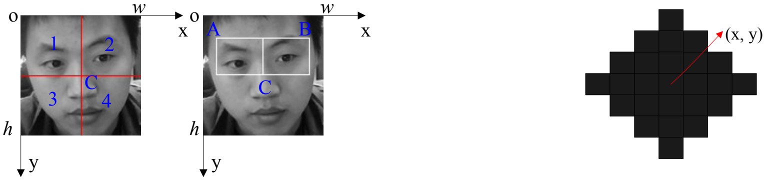


Fig.3. (a) Search area on the face; (b) The shape of eyeball we used.

2) *Search the eyeballs in the determined area*

After the search area is determined, we will search the right eyeball in the left rectangle and the left eyeball in right rectangle. For instance, we search the right eyeball as follows.

First, for one point (x, y) in the left rectangle, add up pixel value of the surrounding points and the sum is denoted as sum(x, y). Then find out the coordinate (i, j) making sum(i, j) be the minimum value.

$$(i,j)=\arg\min_{(x,y)}\{sum(x,y)\}\quad (x,y)\in left\ tectangle \tag{1}$$

$$sum(x,y)=\sum_{n=0}^{n=N} g_n \tag{2}$$

where g_n is the gray value of points around (*x, y*) as shown in the second picture Fig. 3, *N* is total number involved. As the size of images we use is 320×240, the size of faces detected is about 100×100. The rough shape of eyeball is like Fig. 3 (b).

The coordinate (i, j) is the position of the eye in the left rectangle and we can find the other eye in the right rectangle using the same method. Fig.4. shows some examples of result of eye location.

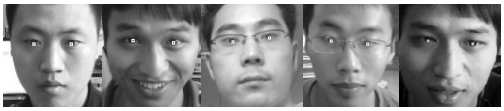


Fig.4. Examples of result of eye location.

3.2. Geometric and illumination normalization

The next step of pre-processing is geometric normalization that contains rotation, cropping, and scaling process based on two eyes. First, rotate the image around the center using bilinear interpolation so that the line through two eyes is horizontal. Second, let the length of line segment E_rE_l be *d* and crop the image to 2d×2d. Last, the face images are scaled to 100×100 using bilinear interpolation.

The images are captured in different background conditions, so they differ greatly in illumination and there are also noises in them. In this paper, histogram equalization is used. This method usually increases the global contrast of many images, and through this adjustment, the intensities can be better distributed on the histogram. Then we use Gaussian filter reducing the noises in the face images. Fig. 5 shows an example.

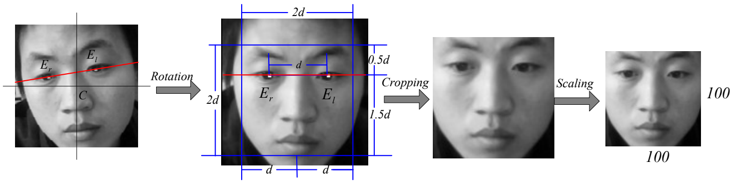


Fig.4. The process of geometric normalization.

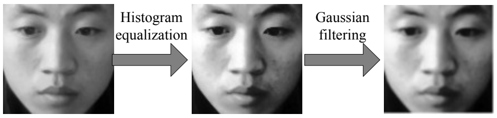


Fig.5. The Histogram equalization and Gaussian filtering.

4. Face recognition using hierarchical multiscale LBP

4.1. Face Recognition Using LBP

The local binary pattern (LBP) [7] was originally proposed for texture description and has achieved promising result in texture classification. In [6], Ahonen proposed a method of face recognition using LBP. Firstly, the face image is divided into R non-overlapping blocks (or regions) of same size. Secondly, every Histogram of LBP codes are calculated over each block. Thirdly, the R histograms are concatenated into a global histogram as the description of the face image. Lastly, a nearest-neighbor classifier is used for face recognition by chi-square distance.

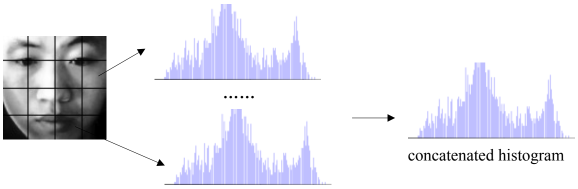


Fig.6. Face recognition using LBP.

4.2. Face Recognition using hierarchical multiscale LBP

The length of a LBP histogram using standard LBP is much longer than the length using LBPu2 (only using uniform pattern). Considering the limited computational resources of mobile phones, LBPu2 is a suitable operator for generating LBP features.

Labeling all “non-uniform” patterns into one pattern makes the feature vector shorter but also means losing more information. Reference [8] presented an improved LBP named hierarchical multiscale LBP. The first picture in Fig. 7 shows an example: the pattern of big radius is non-uniform pattern, but the pattern of small radius is uniform pattern. Thus, we label it to the uniform pattern according to the smaller radius.

First, for each pixel the pattern of the biggest radius is calculated. For those pixels whose pattern is uniform pattern, a sub histogram is built. For those pixels whose pattern is non-uniform pattern, the pattern of smaller radius is further calculated. The process stops when the new pattern of the pixel is uniform pattern or the pattern of the smallest radius is calculated. Finally, all the histograms of different radius are concatenated into one histogram. The second picture in Fig. 7 shows an example calculating LBP from R=4 to R=2.

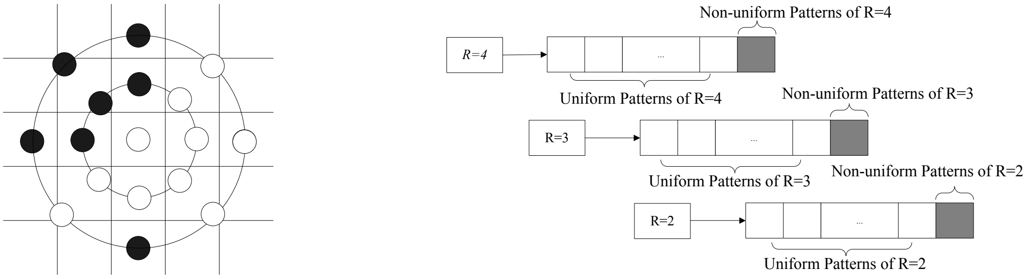


Fig.7. (a) The pattern of big radius is non-uniform pattern, but the pattern of small radius is uniform pattern; (b) An example calculating hierarchical multiscale LBP from R=4 to R=2.

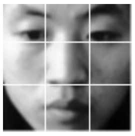
Using hierarchical multiscale LBP for recognition is similar as shown above. First, the image is divided into R blocks of the same size. Then the hierarchical multiscale LBP is calculated for each block. Next, all the hierarchical multiscale LBPs are concatenated into one histogram for recognition. Last, the chi-square distance between histograms is used for classification.

When the image has been divided into blocks, it is expected that some of the blocks contain more useful information such as eyes and mouth. So a weight can be set for each block based on its importance. The weighted D(S, T) statistic becomes:

$$D(S,T)=\sum_{i,j} w_j \frac{(S_{i,j}-T_{i,j})^2}{S_{i,j}+T_{i,j}} \tag{3}$$

where w_j is the weight for block j.

As the limited resolution of images captured from the mobile phone, the image cannot be divided into too many blocks. In this paper, the image is divided into 3×3 blocks as shown in Fig. 8.



4	3	4
2	3	2
1	4	1

Fig.8. A facial image divided into 3×3 blocks and the weights set.

5. Implementation on symbian platform

The LBP based face recognition system is finally implemented on a mobile phone with Symbian OS and an ARM11 processor with 430 MHZ. In our system, only images with size 240×320 are used and only the biggest face is detected during the detection. The processing time of every step is shown in Table 1, from which we know the total time is about 400ms when the number of images enrolled is 100.

We have done some experiments on our system to test the recognition rate of our system. We have taken 300 pictures of 10 people in 3 different dates, each people with 10 pictures in one date. Three experiments are done. In experiment 1, for each people, 4 pictures on first date are randomly selected for enrolment, and all the rest for recognition. In experiment 2, for each people, 4 pictures on first date and 4 pictures on second date are randomly selected for enrolment. In experiment 3, for each people, 4 pictures on each date are randomly selected for enrolment. The recognition rate is shown in table 2.

Table 1. Processing time of every step, where N is the number of images enrolled before recognition.

Step	Face Detection	Pre-processing	Feature Extraction	Classification
Processing Time(ms)	255	83	12	12 + 0.5N

Table 2. The recognition rate of different experiment.

Experiment	Experiment 1	Experiment 2	Experiment 3	Average Rate
Recognition Rate	82.31%	87.72%	93.33%	87.78 %

6. Conclusion

This paper introduced an implementation a LBP based fast face recognition system on Symbian platform in detail. The AdaBoost algorithm is used for face detection. During face preprocessing, a rapid eye location method is proposed and implemented. In the last, improved LBP is adopted for recognition. Eventually, the system can process about 2.5 frames per second and its average recognition rate is up to 87.78 %.

Acknowledgements

This work is supported by the Industry, Education and Research Project of Guangdong Province (No. 20100901).

References

[1] K. Venkataramani, S. Qidwai, B. VijayaKumar. Face authentication from cell phone camera images with illumination and temporal variations. IEEE Trans. on Systems, 2005, vol. 35, no. 3, 411–418.
[2] Shih-Wei Chu, Mei-Chen Yeh, Kwang-Ting Cheng. A Real-time, Embedded Face-Annotation System. MM'08, Vancouver, British Columbia, Canada. 2008, October 26–31.
[3] S. Z. Li, A. K. Jain. Handbook of Face Recognition. Springer-Verlag. New York; 2005.
[4] Ming-Hsuan Yang, David Kriegman, Narendra Ahuja. Detecting faces in images: A survey. Pattern Analysis & Machine Intelligence 24(1), January 2002, 34-58.
[5] E. Hjelmas, B. K. Low. Face detection: A survey. Computer Vision and Image Understanding, 2001;83:236-274.
[6] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: Application to face recognition, Transactions on Pattern Analysis and Machine Intelligence, 2006; vol. 28, no. 12, 2037–2041.
[7] Ojala, T., Pietik¨ ainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29 (1996) 51–59.
[8] Zhenhua Guo, Lei Zhang, David Zhang, and Xuanqin Mou, "Hierarchical multiscale LBP for face and palmprint recognition", Image Processing (ICIP), 2010; 4521-4524.