International Journal of Technology (2014) 2: 159-168
ISSN 2086-9614

FACE RECOGNITION USING AVERAGE HALF FACE TEMPLATE

Muhammad Imran Shehzad[*1], Muhammad Awais[1], Mohsin Amin[1], Yasir Ali Shah[1]

[1]*Electrical Engineering Department, COMSATS Institute of Information Technology, University Road, Tobe Camp Postal Code 22060 Abbottabad, Pakistan*

(Received: October 2013 / Revised: June 2014 / Accepted: June 2014)

ABSTRACT

Face recognition is one of the most important technologies, which has been well-developed over the last two decades. Face recognition technology has reached a level of utmost importance as the security issues increase worldwide. Most of the previously proposed systems, based on half face images are computationally slow and require more storage. In the proposed model, an average half face image is used for recognition to reduce computational time and storage requirements. The Viola Jones method is used in conjunction with intensity-based registration for real time face detection and registration, before splitting the full face. Finally, Principal Component Analysis (PCA) is used to compress the multi-dimensional data space and recognition. Experimental results clearly elaborate that half face recognition produces much better results as compared to the full face recognition and other previously proposed half face recognition models.

Keywords: Ada Boost (Ad Boost); Face Registration; Principle Component Analysis (PCA)

1. INTRODUCTION

In the recent years, there is lot of work done in the field of biometric technology. Biometric technology changes the concept of security; it has given more depth to the investigations in order to recognize the precise identity of individuals. One can even identify the individuals in a group that are under surveillance. It uses human biographical features like face, ears, eyes and thumb for security purposes for user verification and identification. Face recognition is also a sub-branch of Visual Biometrics and is the most widely used technique for recognition. Most of the proposed face recognition models have used a holistic approach, only a few researchers have addressed dealing with an arbitrary patch or half face for recognition (Shengcai Liao et al., 2013). However, all these models are computationally expensive and require more storage. Consequently newer research is more focused on systems that are computationally inexpensive and requires the least storage. Therefore, now researchers are more interested to use face symmetry for recognition, as there is no need to process or store the other half as it is exactly the same for most of the cases.

Ramanathan et al. (2004) are the pioneers in discussing the use of the "Half Face". Selection of the left or the right half face has shown some interesting results, the accuracy of the recognition models, when using one-half of the face has approached that of the full face. Harguess et al. (2008) have presented the idea of using an average half face; they have achieved much better results as compared to the full face. Gnanaprakasam et al. (2010) have come up with an idea of using wavelets with the half face to reduce storage requirements and computational cost.

[*] Corresponding author's email: imranshahzad@ciit.net.pk, Tel. +92-992-383591-6 Ext.326, Fax. +92-992-3834441
Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v5i2.408

Sharma and Vashisht (2012) has used "Elastic Bunch Graph Matching" with an average half face for recognition purposes to further reduce computational cost. Face symmetry also has been exploited by the researchers for 3D face recognition to reduce computational and storage overhead. Few researchers have used facial symmetry to handle pose variations in real time and in an uncooperative image acquisition environment (Passalis et al., 2011). The limitations of the above mentioned methods are that these are still computationally quite expensive and slow, not in real-time, semi-automated, tested on few standard databases and their accuracy rate is not high.

In the paper, we have proposed a system which is more robust, fully automated, and computationally inexpensive and achieves quite a high degree of accuracy.

2. METHODOLOGY

The proposed design has been divided into the following modules as shown in Figure 1.

2.1 Face Detection

2.2 Face Registration

2.3 Face Splitting (Average Half Face)

2.4 Face Recognition

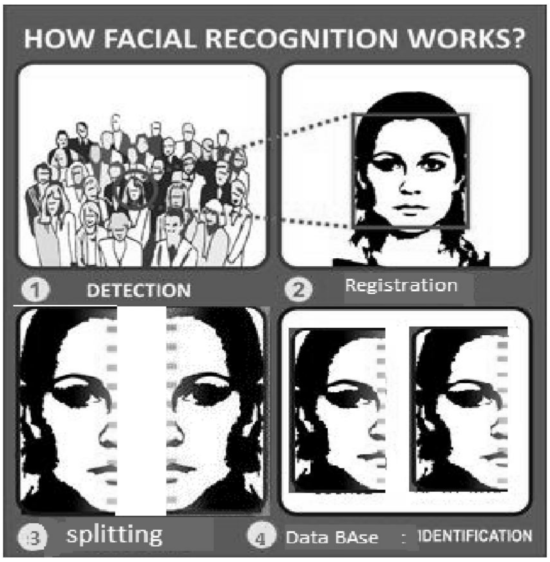

Figure 1 Design flow

There are many methods available for face detection like the skin tone method (Yao & Gao, 2001), support vector machines (SVM) (Jee et al., 2004), the neural networks method (Tsai et al., 2006) and the Voila-Jones method (Viola & Jones, 2001). We have used the Viola-Jones method for face detection. The Viola-Jones method is one of the most prominent methods used for face detection. The detection rate and computation time of the Voila-Jones method is much faster than other algorithms and it gives excellent results. In the 2nd phase of the paper, the detected face is registered for the pose correction. In the 3rd phase, the face is split into two equal parts to discard the redundant information. In the 4th and final phase of the paper, face recognition is performed using PCA. It is the most widely used method for face recognition; we have used it for half faces instead of a full face.

The proposed system is tested on the Essex frontal image database for a controlled environment; we made our own database to test our algorithm on an uncontrolled environment and used some random database of other universities to ensure the efficiency of proposed system.

2.1. Face detection

We have used the Voila-Jones method for face detection. It involves four key concepts: the Haar-like feature usually known as rectangle features, for a faster mode of the detection that used an integral image, a machine learning method called Ada Boost, and a cascade classifier in series to combine features efficiently.

There are thousands of Haar-like features in the human face, we used only three main features for face detection, nose, eyes and lips (See Figure 2); however, the computations take more time because it has to sum up the pixel values of all the pixels until the point of the window of rectangle features the value I.

$$I(x,y)=\sum_{\substack{x'\le x\\ y'\le y}} i(x',y')$$

$$I=\sum(Pixels\_white\_area)-\sum(Pixels\_black\_area)$$

$$If\,(I\ge Threshold)$$
$$(Face)$$
$$Else$$
$$(Non\_Face)$$

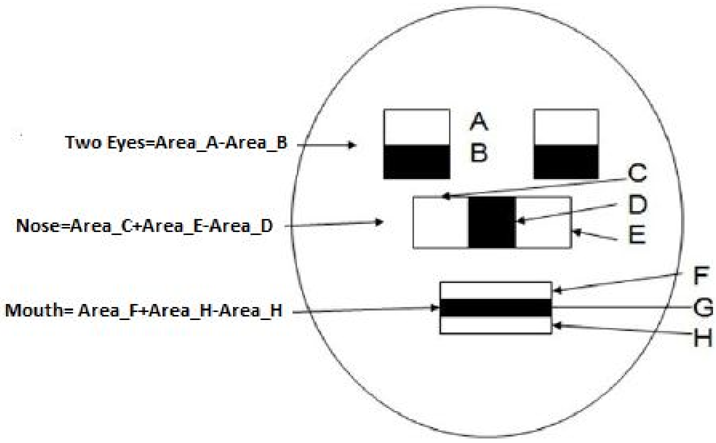

Figure 2 Haar-like features

In order to make it computationally fast, an integral image is used at any point in an image which carries the information of all the pixels till that point. So, we do not need to slide the window again and again which reduces computational time as compared to the previous method (See Figure 3).

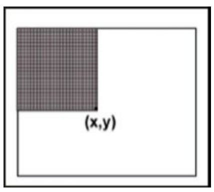

Figure 3 Integral image

To select the specific Haar-like feature and to set the values of the threshold we use a machine learning method normally known as Ada-Boost. It combines many weak classifiers to make them strong classifiers. Ada Boost selects the weak classifiers and assigns a weight respectively to each classifier. Finally the last resultant weighted combination is the required strongest classifier. Each strong classifier has its own threshold value. Now the last step in face detection is to combine a series of classifiers (See Figure 4).

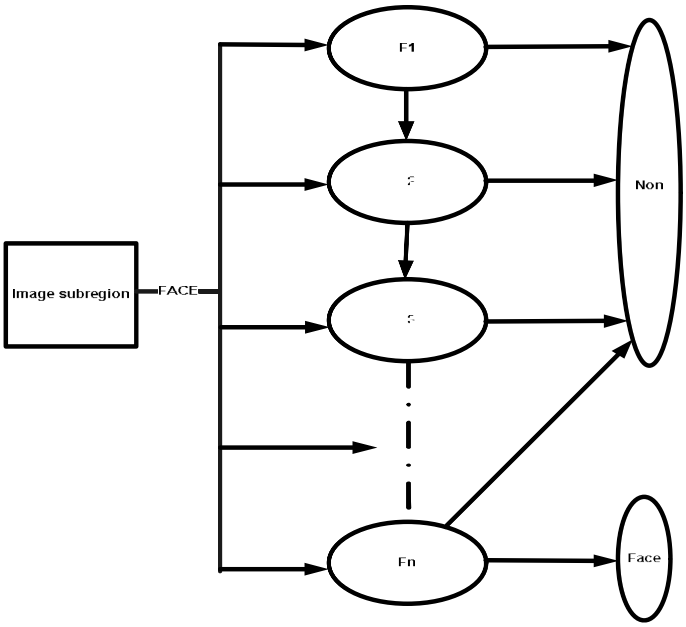

Figure 4 Ada-boost classifiers

Some results produced by this method are shown below (See Figure 5).

Figure 5 Face detection results

2.2. Face registration

Face registration is very critical as we are dealing with the half face. In order to cut the face properly, without losing significant information, it is necessary to discard only the redundant information. We use an intensity-based image registration method (Rohde, 2008), as our detection system is quite good. The intensity-based registration method helps us to align the images that are slightly tilted with less computation time. This method works on the following principles: the process begins with the transformation (Rigid Transformation) and an initial transmission matrix. Bi-linear Interpolation is used to determine image transformation that is applied to the moving image. Next, the Metric compares the transformed moving image to the fixed image to compute the metric value. Finally, the optimizer checks for a stop condition. The complete process flow is shown in Figure 6.

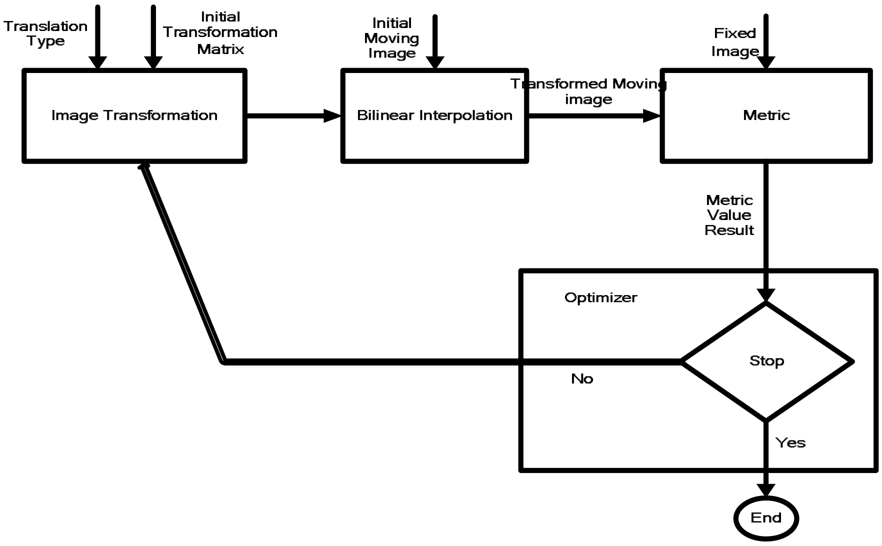

Figure 6 Registration flow

We are dealing mostly with frontal faces, so we use a rigid transformation, because the faces can be slightly tilted or move a little. In Bi-linear interpolation, we create the dummy metric using some reference face to align the source image to the standard reference face selected manually. This is done in two steps of rotation and translation.

I. Rotation

$$X = VCOS\theta - WSIN\theta$$
$$Y = VSIN\theta + WCOS\theta$$

II. Translation

$$X = V + t_x$$
$$Y = W + t_y$$

(V, W) → Transformed image pixel
(X, Y) → Original image pixel

We take the square mean of the transformed image and the reference image (See Figure 7), and then the optimizer checks the value to see whether it is above or less than the threshold. If it is less than the threshold registration process at the end, otherwise it repeats again and again until it reaches the threshold.

Figure 7 Reference image

Some results of registration process are shown below (Figure 8).

Figure 8 Registration results

2.3. Face splitting

Splitting is quite simple after registration is done properly. The vector split method is used to split the face into two halves and save the average half face for further processing and discard the redundant information present in the full face. Before discarding the half part it is compared with the other part by using Euclidean distance. If the difference is more than the threshold and more than save the distance as well, this will be helpful for face recognition. Some results of face splitting are shown below (Figure 9).

Figure 9 Face splitting results (Average Half Face)

2.4. Face recognition

For the face recognition procedure, we used PCA. It is a statistical method widely used in image recognition and compression. We used PCA in such a manner so as to find the data patterns in high dimensional data space. Patterns are analyzed by applying different statistical tools on the data, and thus features of face are calculated. Once the features are calculated, then it is easy to recognize the face in a given database. The size and shape of human nose, eyes, lips etc. can be considered as features. Wearing glasses, a hat or other visual noise in an image, or the rotation of the face can distort the information about the features. (Zhu & Cutu, 2013).

2.4.1. Working Principle of PCA in half face

The database of images is arranged in the form of a matrix, such that each column of that matrix represents an image. Let us consider that we have images of the size MxN and there are a total P number of images in the database, then we have a matrix X of size $MNxP$. The following steps will be followed in order to create feature vectors, Eigen faces and finally to recognize faces.

2.4.2. Mean & mean shifted images

Mean (m) of each image is calculated as:

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

And then the images are mean-centred by subtracting the mean image from each and every image vector. If the mean-centred image is defined as w_i then

$$w_{i=x_i-m} \tag{2}$$

$$X = [w_1 w_2 w_3 \ldots\ldots\ldots\ldots w_P] \tag{3}$$

2.4.3. Co-variance

Then the next step is to calculate the co-variance matrix by using the formula for co-variance.

$$A = Cov(X) \tag{4}$$

Now a matrix L is constructed, by multiplying the transpose of A by a simple matrix A.

$$L = A'A \quad ; \qquad (L_{PxP}) \tag{5}$$

Here, L is a PxP matrix, obtained by multiplying a $PxMN$ matrix with a $MNxP$ matrix. This intelligent multiplication results in a smaller matrix to be dealt with, especially if we reverse the order of multiplication, then a very large matrix of size $MNxMN$ would be the result; and it would be difficult to deal with a matrix of that great a size.

2.4.4. Eigenvector & Eigenvalue

All face images in the database can be represented as a linear combination of the Eigen faces. So the next step is to calculate the Eigen values and the Eigen vectors as:

$$VL=\lambda L \tag{6}$$

2.4.5. Principal Components

Only the top 50% of the values are selected; and are retained. Thus a matrix $B_{Px\frac{P}{2}}$ is obtained. When this Matrix B is multiplied with Matrix A, the principal components are obtained as:

$$PC = B'X; \quad (\tfrac{P}{2}xPxMNxP) \tag{7}$$

Here, PC is $\frac{P}{2}xP$ matrix.

2.4.6. Calculate similarity

Now, the main step of recognition is checking the similarity between two images (input image and database image). Consequently, the Euclidean distance between the two images is calculated, and the image in the database having the minimum difference with the input test image is either selected or recognized. Steps for calculating the similarity between images are:

$$TESTimage = InputImage - m; \ (MN \times 1) \tag{8}$$

$$t = PC'TESTimage; \ (\tfrac{P}{2} \times P \times MN \times 1) \tag{9}$$

$t \ is \ (\frac{p}{2} \times 1)$

The Euclidean distance is calculated by using the following formula:

$$Euc - dis = \sqrt{\sum_{i=1}^{n}(PC_i - t)^2} \quad (10)$$

The threshold is set statistically, whether the image is present in database or not. The threshold is chosen very carefully after scanning the 182 faces individually for the complete database to estimate the full range of the threshold. We also used a cosine distance to check how the accuracy and the threshold are set, as explained above.

3. RESULTS AND DISCUSSION

Few of the true matching results have been shown in Figure 10, where left face image is the test image and the right face is recognized from the database. It can be observed from the results, that our proposed system has achieved recognition efficiently irrespective of pose, illumination and feature's variations. Figure 11 shows few of the false matching results as well. Whereas, Table 1 shows the accuracy of our proposed model, this model is much better in terms of recognition as compared to the earlier systems (Sharma & Vashisht, 2012; Satone & Kharate, 2012).

Figure 10 True matching results

Figure 11 False matching results

This model works quite well for half face images as compared to the full face images. The overall results are summed up in Tables 1 and 2.

Table 1 Recognition results

Section	Total images(Essex Database, Real time images)	True Results	Accuracy (%)
Half Face	182	175	96.15%
Full Face	182	168	92.00%

Table 2 Computational Time

Section	Computational Time
Half Face	2.84 Seconds
Full Face	4.54 Seconds

4. CONCLUSION

This paper proposes a novel algorithm for face recognition using an average half face, to make it real time and robust. The proper detection and registration is performed before splitting the face in two halves, finally PCA is used for recognition. We have tested the system for an uncontrolled environment and a real time environment as well. The results reveal that this system is more robust when compared to the other systems. Its real time application makes it more beneficial for surveillance purposes. It takes less than 3 seconds to recognise a person by using MATLAB and its computational time can be further reduced by using other platforms.

5. ACKNOWLEDGEMENT

The authors are grateful to Mr. Zahid for Technical support in MATLAB and Mr. Baseer for language help. We also want to thank reviewers for their valuable comments

6. REFERENCES

Gnanaprakasam, C., Sumathi, S., RaniHema Malini, R., 2010. Average Half Face in 2D and 3D Using Wavelets for Face Recognition. *Proceedings of the 9th WSEAS International Conference on Signal Processing*

Harguess, J., et al., 2008. 3D Face Recognition with the Average Half Face. *International Conference on Pattern Recognition ICPR*, pp. 1–4

Jee, H., Lee, K., Pan, S., 2004. Eye and Face Detection using SVM. *J. IEEE Intelligent Sensors, Sensor Networks and Information Processing Conference,* Volume 10, pp. 577–580

Passalis, G., Perakis, P., Theoharis, T., Kakadiaris, I.A., 2011. Using Facial Symmetry to Handle Pose Variations in Real–world 3D Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 33(10), pp. 1938–1951

Ramanathan, N., Chowdhury, A.K.R., Chellappa, R., 2004. Facial Similarity Across Age, Disguise, Illumination and Pose. *IEEE International Conference in Image Processing (ICIP),* Volume 3, pp. 1999–2002

Rohde, G.K, 2008. Intensity–based Image Registration. *Introduction to Biomedical Imaging and Image Analysis*. pp. 42–431

Satone, M.P., Kharate, G.K., 2012. Face Recognition Based on PCA on Wavelet Subband of Average Half face. *Journal of Information Processing Systems,* Volume 8(3)

Sharma, V., Vashisht, R., 2012. Average Half Face Recognition by Elastic Bunch Graph Matching based on Distance Measurement. *International Journal for Science and Emerging Technologies with Latest Trends,* Volume 3(1), pp. 24–35

Shengcai L., Jain, A.K., Li, S.Z., 2013. Partial Face Recognition: Alignment–free Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 35(5), pp. 1193–1205

Tsai, C.C., Cheng, W.C., Taur, J.S., Tao, C.W., 2006. Face Detection Using Eigen Face and Neural Network. *J. IEEE International Conference on System, Man, and Cybernetics,* Volume 10, pp. 4343–4347

Viola, P., Jones, M., 2001. Robust Real–time Object Detection. *Second International workshop on Statistical and Computational Theories of Vision–Modelling, Learning, Computing, and Sampling*. Vancouver, Canada. July 13, 2001

Yao, H., Gao, W., 2001. Face Detection and Location based on Skin Chrominance and Lip Chrominance Transformation from Colour Images. *J. Pattern Recognition,* Volume 34, pp. 1555–1564

Zhu, Y., Cutu, F., 2013. Face Detection Using Half Face Templates. *Journal of Vision,* Volume 3(9), pp. 839