A Fast Mobile Face Recognition System for Android OS Based on Eigenfaces Decomposition

Charalampos Doukas[1] and Ilias Maglogiannis[2]

[1] University of the Aegean, Samos, Greece
doukas@aegean.gr
[2]University of Central Greece, Lamia, Greece
imaglo@ucg.gr

Abstract. This paper presents a speed-optimized face recognition system designed for mobile devices. Such applications may be used in the context of pervasive and assistive computing for the support of elderly suffering from dementia in recognizing persons or for the development of cognitive memory games Eigenfaces decomposition and Mahalanobis distance calculation have been utilized whereas the recognition application has been developed for Android OS. The initial implementation and the corresponding results have proven the feasibility and value of the proposed system.

Keywords: Mobile face recognition, Android OS, Face Detection, Eigenfaces decomposition.

1 Introduction

Face recognition refers to automatically identifying or verifying a person from a digital image or a video frame from a video source. The process involves the face detection in images or image sequences, proper feature extraction, the creation of a reference set based on known faces and the detection of new faces based on the latter. Mobile face recognition can be utilized either for user authentication or for cognitive assistance in identifying individuals. The latter can be particularly useful for the elderly whose face recognition ability declines while aging. A plethora of face recognition algorithms have been proposed in literature ([1] – [12]). However, the application of the well known and established techniques to mobile devices, such as mobile phones, raises some challenges related to two major issues: the inefficiency of the algorithms and the limited processing capabilities of the mobile devices.

This paper presents a speed-optimized method for performing face recognition on Android mobile phones. Both training and recognition phases are performed on the device. The Android mobile operating system provides built-in functionality for detecting faces on still images. Eigenface decomposition is used in order to extract important features from the images and a training vector is created based on an initial image training set. The Mahalanobis cosine distance is utilized for measuring the similarity of the given train set images and the examined image in the Eigenface projection space. The latter methods have been selected for implementation on a mobile device due to lower complexity and memory requirements compared to other methods

H. Papadopoulos, A.S. Andreou, and M. Bramer (Eds.): AIAI 2010, IFIP AICT 339, pp. 295–302, 2010.

available in literature. In addition to the face recognition process performed thoroughly on the device, a web service has been developed that allows the feature extraction and classification using more advanced dace recognition techniques. The latter method requires network connectivity from the mobile device but can provide more accurate results.

Initial evaluation of the system has been performed on Google's G1 mobile phone and results are quite promising regarding the system's performance and accuracy. The paper presents the aforementioned, discusses related work and includes technical information and implementation details.

2 Related Work and Background Information

In general, the whole process of face recognition consists of detecting and selecting faces from the current image or video frame utilizing techniques like skin detection, process the selected image region and finally applying a recognition algorithm.

Several classifiers have been proposed in the literature e.g. minimum distance classification in the eigenspace ([11], [12]), Fisher's discriminant analysis [3], and neural networks [2]. Global techniques work well for classifying frontal views of faces, however they are not robust against pose changes since global features are highly sensitive to translation and rotation of the face. To avoid this problem an alignment stage can be added before classifying the face. Aligning an input face image with a reference face image requires computing correspondence between the two face images. The correspondence is usually determined for a small number of prominent points in the face like the center of the eye, the nostrils, or the corners of the mouth. The image region is also converted to grey levels and equalized, then cropped with an elliptic mask and normalized. Finally, the recognition algorithm compares the processed image to a set of prerecorded faces, all belonging to the same subject, to find the best match. This algorithm usually returns also a measure of similarity between the current face and the reference template.

2.1 Face Detection and Preprocessing

One of the most crucial parts in the presented face recognition system is the preprocessing of the considered image region. To align a face horizontally, a common technique consists in locating the eyes and then rotating the selected image so that their inclination is null. Also, the distance between the eyes is used to resize the same image to a given value of width and height.

Once the positions (uR, vR) and (uL, vL), of the right and left eye respectively, have been determined, we simply calculate the angle and the distance between them as follows:

$$a_{RL} = \tan^{-1}\left(\frac{vL - vR}{uL - uR}\right) \tag{1}$$

$$d_{RL} = \sqrt{(u_L - u_R)^2 + (v_L - v_R)^2} \tag{2}$$

The face is then rotated of an angle $-a_{RL}$ in order to align the eyes, and then resized to 24x24 pixels, i.e. the size of the templates used for comparison. Rotation and scaling, centered on the right eye (u_R, v_R), are performed with a simple affine transformation as follows:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} a & b & (1-a)u_R - bv_R \\ -b & a & bu_R + (1-a)v_R \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

$$a = s\cos(-a_{RL})$$
$$b = s\sin(-a_{RL})$$

where (u, v) are the pixel coordinates of the source image and (u′, v′) those of the destination. In order to avoid possible outliers in the final image after rotation, the affine transformation is actually applied to a sub-region slightly bigger than the original face bounding box. The final step of the image processing consists in cropping the face's area with an elliptical mask. This reduces the influence of hair and background pixels at the four corners of the rectangular region.

2.2 Eigenfaces Decomposition and Similarity Detection

After the image processing described above, the most popular techniques for face recognition, called Eigenfaces [12] is applied. Eigenfaces decomposition and similarity detection relies on measuring the similarity between a new face image and a reference one, projecting both the images into an eigenspace, previously created by training, and calculating the distance between the projections. The idea behind the eigenface technique is to extract the relevant information contained in a facial image and represent it as efficiently as possible. Rather than manipulating and comparing faces directly, one manipulates and compares their representations.

To create a set of eigenfaces, one must prepare a training set of face images. The pictures constituting the training set should have been taken under the same lighting conditions, and must be normalized to have the eyes and mouths aligned across all images. They must also be all resampled to the same pixel resolution. Each image is treated as one vector, simply by concatenating the rows of pixels in the original image, resulting in a single row with r × c elements. For this implementation, it is assumed that all images of the training set are stored in a single matrix T, where each row of the matrix is an image. The mean is then subtracted. The average image *a* has to be calculated and then subtracted from each original image in *T*. The eigenvectors and eigenvalues of the covariance matrix *S* are thereafter calculated. Each eigenvector has the same dimensionality (number of components) as the original images, and thus can itself be seen as an image. The eigenvectors of this covariance matrix are therefore called eigenfaces (see Fig. 1). They are the directions in which the images differ from the mean image. The *D x D* covariance matrix will result in *D* eigenvectors, each representing a direction in the $r \times c$ dimensional image space. The eigenvectors (eigenfaces) with largest associated eigenvalue are kept. The latter can now be used to represent both existing and new faces: we can project a new (mean-subtracted) image

on the eigenfaces and thereby record how that new face differs from the mean face. The eigenvalues associated with each eigenface represent how much the images in the training set vary from the mean image in that direction.

The computation of eigenvectors is performed as follows:

Let *T* be the matrix of preprocessed training examples, where each row contains one mean-subtracted image. The covariance matrix can then be computed as $S = \mathrm{T}^{\mathrm{T}}\,\mathrm{T}$ and the eigenvector decomposition of *S* is given by:

$$Sv_i = T^T T v_i = \lambda_i v_i \tag{4}$$

However TTT is a large matrix, and if instead we take the eigenvalue decomposition of:

$$TT^T u_i = \lambda_i u_i \tag{5}$$

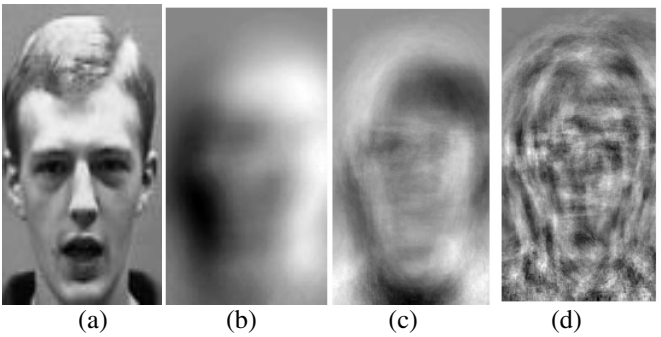
then we notice that by pre-multiplying both sides of the equation with TT, we obtain:

$$T^T TT^T ui = \lambda_i T^T u_i \tag{6}$$

Meaning that, if $\mathbf{u}_i$ is an eigenvector of $\mathbf{TT}^{\mathrm{T}}$, then $\mathbf{v}_i=\mathbf{T}^{\mathrm{T}}\mathbf{u}_i$ is an eigenvector of *S*.

Regarding measuring the similarity between different eigenvectors and performing image recognition, several methods have been proposed. The most common one, which is considered to give the best performance, is the Mahalanobis Cosine distance ([13], [14]). As the name suggests, this is given by the cosine of the angle θij between the two face projections *f*i and *f*j in the Mahalanobis space:

$$\xi(fi, fj) = -\frac{fi \bullet fj}{\|fi\| \|fj\|} = -\cos\theta ij \tag{7}$$



(a) (b) (c) (d)

Fig. 1. Example illustrating eigenfaces decomposition: a) original face image, b) first eigenface, c) tenth eigenface, d) hundredth eigenface.

3 Technical Details and System Architecture

The mobile face recognition system has been developed utilizing Google's Android mobile Operating System (OS) [15] and the appropriate software development kit (sdk).

Fig. 2. Architecture of the proposed mobile face recognition system

Android is a mobile operating system running on the Linux kernel. Several mobile device vendors already support it. The platform is adaptable to larger and traditional smart phone layouts and supports a variety of connectivity technologies (CDMA, EV-DO, UMTS, Bluetooth, and Wi-Fi). It supports a great variety of audio, video and still image format, making it suitable for capturing and displaying images. Finally, it supports native multi-touch technology, which increases the application's usability.

Fig. 2 presents an illustration of the proposed system architecture. The mobile application utilizes native Android resources for capturing images using the mobile devices' image recording capabilities. The images can also be allocated on a remote or local repository. Android also provides native face detection functionality. Appropriate java methods can be used directly in any custom application and allow the detection of faces within images. The methods provide the coordinates of the eyes. Appropriate java classes (referred as modules) have been developed for performing the initial image processing (i.e., face cropping, rotation and scaling), the decomposition of the initial image into eigenfaces and the projection of the eigenvactors into the Mahalanobis domain and distance calculation between a given new image instance and the ones used as a training set. The images and the computed eigenvectors of the training set reside within the mobile device's storage repository (e.g., a memory card).

In addition to the face recognition process implemented locally on the mobile device, a web service has bee also developed that enables a more advanced feature extraction and classification of the face images. An appropriate module enables the communication between the Android application and the Web Service utilizing SOAP messages. The Web Service has been developed using Java and Apache Axis [17]. The classification module is based on the same java code developed for the Android for pre-processing the images and extracting the eigenvectors. Classification of the latter is performed using the Weka tool [18] that contains a great variety of tools for pattern recognition. The data exchange between the mobile device and the web service involves the exchange of images and the classification results.

Fig. 3 illustrates the main steps for performing face recognition on an Android-enabled mobile device. When faces are detected in an image, a validation of the faces found and the corresponding eye coordinates and eye distances is performed in order to minimize erroneous cases.

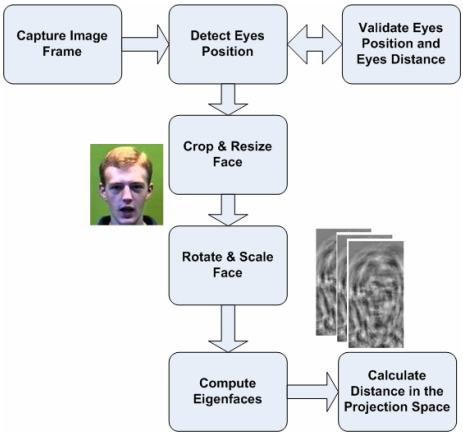


Fig. 3. Data flow diagram illustrating the basic steps for face detection and recognition using eigenface decomposition

4 Initial Evaluation Results

In order to evaluate the proposed system a HTC G1 [16] mobile phone running Android OS version 1.6 has been used in order to host the developed face recognition application. The initial training set consisted of 126 images containing 7 different faces (4 males and 3 males, 18 sample images per face). All faces have been cropped from the background as described in Section 2 and resized to a 100x100 pixels resolution. All images reside into the mobile phone's memory card and the developed face recognition application has been initially used in order to generate the corresponding eigenfaces. The first 50 eigenvalues have been for each image and the corresponding vectors have been saved into the mobile's phone memory card. For the evaluation, 5 different images (i.e. not included in the training set) from each individual have been used.

The creation of the training set has taken an average of 65 seconds for all 126 images on the device, whereas the processing, eigenfaces decomposition and mahalanobis distance calculation and comparison for each evaluation image has taken approximately only 1.2 seconds. The overall system accuracy in detecting faces in images is about 96%, whereas the total accuracy in recognizing faces that do not belong to the training set is 80%. The latter are quite promising considering the size of the evaluation set and the fact that both the training and recognition phases are performed on the mobile device at very reasonable time delays, allowing the usage of the proposed face recognition system in real applications.

Regarding the ability of the system to utilize more advanced classifiers through the Weka tool for more accurate face recognition, a second experiment has been conducted. The same training set has been used, whereas for the evaluation process, the latter set along with new image samples have been used through 10-fold cross validation. Several classifiers like Bayes Networks, Support Vector Machines (SVM), and Neural Networks (NN) have been validated. The best performance has been achieved by SVM approaching a total accuracy of 98%, followed by 96.2% achieved by NN and 95% by the Bayes classifier respectively. The total time to process and classify an

Table 1. Initial evaluation results for the proposed mobile face recognition system. System accuracy in face detection within images and overall face recognition accuracy are presented. Both training and recognition phases have been performed on the mobile device.

	Accuracy in detecting faces in images in training set (%)	Accuracy in Face Recognition (%)
Male 1	94.4	80
Male 2	100.0	80
Male 3	100.0	100
Male 4	88.8	60
Female 1	88.8	80
Female 2	100.0	80
Female 3	100.0	80

image is less than 1 sec at this case, but the overall network transmission of data introduces further delay until the result is displayed at the mobile device.

5 Conclusions

The advances of state of the art smart phones and PDAs have enabled the creation of fast and reasonably robust face detection/recognition applications like the one presented in this work. Such applications may be used in the context of pervasive and assistive computing for the support of elderly suffering from dementia in recognizing persons or for the development of cognitive memory games.

An important issue to be handled in face recognition is the "Face Variation" problem. This is closely related to changes of illumination, pose, and expression. As non uniform illumination affects the acquired face image, it is important to know how the recognition algorithm works under varying illumination. Pose variations include approaches on how to model faces seen from different points of view. Expression variations refer to how we can design algorithms for the recognition of emotions and other facial expressions [10]. Future research will focus on addressing such issues, on improvement of detection precision, the addition of online training module and other novel applications.

References

1. Chellappa, R., Wilson, C.L., Sirohey, C.: Humain and machine recognition of faces: A survey. Proc. IEEE 83(5), 705–740 (1995)
2. Fleming, M., Cottrell, G.: Categorization of faces using unsupervised feature extraction. In: Proc. IEEE IJCNN International Joint Conference on Neural Networks, pp. 65–70 (1990)
3. Belhumeur, P., Hespanha, P., Kriegman, D.: Eigenfaces vs fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)

4. Jonsson, K., Matas, J., Kittler, J., Li, Y.: Learning support vectors for face verification and recognition. In: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, pp. 208–213 (2000)
5. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(10), 1042–1052 (1993)
6. Tolba, A.S., El-Baz, A.H., El-Harby, A.A.: Face Recognition: A Literature Review. International Journal of Signal Processing 2(2) (2006)
7. Hong, L., Jain, A.: Integrating faces and fingerprints for personal identification. IEEE Trans. Pattern Analysis and Machine Intelligence 20(12), 1295–1307 (1998)
8. Deniz, O., Castrillon, M., Hernandez, M.: Face recognition using independent component analysis and support vector machines. Pattern Recognition Letters 24, 2153–2157 (2003)
9. Li, Y., Gong, S., Liddell, H.: Support vector regression and classification based multi-view face detection and recognition. In: Proc. IEEE International Conference on Face and Gesture Recognition, Grenoble, France (March 2000)
10. Maglogiannis, I., Vouyioukas, D., Aggelopoulos, C.: Face Detection and Recognition of Human Emotion Using Markov Random Fields. In: Personal and Ubiquitous Computing. Springer, Heidelberg, doi: 10.1007/s00779-007-0165-0
11. Sirovitch, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A 2, 519–524 (1987)
12. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 72–86 (1991)
13. Aly, S., Tsuruta, N., Taniguchi, R.-I.: Face recognition under varying illumination using Mahalanobis self-organizing map. Journal of Artificial Life and Robotics 13(1), 298–301 (2008)
14. Beveridge, R., Bolme, D., Teixeira, M., Draper, B.: The CSU Face Identification Evaluation System User's Guide: Version 5.0, Computer Science Department, Colorado State University (May 2003)
15. The Android mobile OS by GoogleTM, `http://www.android.com/`
16. HTC G1 mobile phone, `http://www.htc.com/www/product/g1/overview.html`
17. The Apache Axis Framework, `http://ws.apache.org/axis/`
18. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)