Novel Face Recognition System for Blind People

Chilaron M., Dunai L.
Research Center in Graphic Technology
Universitat Politècnica de València,
Valencia, Spain
ladu@upv.es

***Abstract* — This paper describes the face detection and recognition system for blind people in real time. Two main aspects on machine learning compose the work: the object detection based on boosted cascade and the face recognition based on Eigenfaces. The system is developed on Raspberry hardware.**

***Keywords* — object detection; face recognition; boodting cascade; simple features; eigenfaces**

I. Introduction

Vision is one of the main perception senses of the human sensorial mechanism beside sense of smell, sense of hearing and feeling [2]. Loss of vision prevents humans to develop autonomously their daily activities [1]. The inability to recognize or identify people during the stay at home or during meetings become a frustration (inconvenient) for blind people and mainly when existing in the area people are quite and does not make noises.

Speech, object and face recognition systems have a great impact on the wearable systems for blind people. They may convey humans form the near environment into speech. Through face detection and recognition systems, blind people or people with low vision may perceive persons from the environment, familiars, friends or colleagues at work. Blind people may be informed about the presence of people in near environment and identify them.

There are many applications for real-time object detection [3] face recognition [4], text recognition [5] and currency bills identification [6, 7]. Each work makes emphasis on a unique methodology and algorithms. For example, on [3], the object detection uses cascade structure and AdaBoost classifiers based on Haar basis functions. On [4] uses Eigenfaces based on the Turk and Pentland models [8, 9]. [10] Implement the Principle Component Analysis or Eigenfaces for face recognition in order to perceive facial expressions, emotions and gesture. PCA and was used in [7] for dollar bill recognition.

The work is focused on developing a face detection and face recognition algorithms brought together for a system for blind people. The face detection used the implementation of the algorithms proposed by Viola and Jones, which consist on the frontal face detection. Up to date this method is the most used in the area of pattern recognition (Viola & Jones, 2001). In order to achieve the objective the Haar Cascades functions were used. The face recognition uses the principal Component Analysis of the Eigenfaces algorithms. The system works on a raspberry pi hardware and smartphones. Initially the system was developed for raspberry pi hardware due to its capacity of processing and its peripheral webcam, having a free open code, also because the hardware could use the OpenCV libraries and C++ bindings for Python. Therefore the raspberry pi camera has the 5 megapixel and Bluetooth connection.

II. System Description

Using Model B Raspberry Pi computer unit and raspberry Pi camera developed the system. The Model B Raspberry Pi has 3,5W, an ARM1176JZF-S 700MHz processor, 512 Megabitess of RAM, VideoCore IV GPU and multiple Input/Output interfaces.

The Raspberry Pi of a credit-card size computer runs under Python and having access to OpenCV libraries.

III. Face Detection

A. Face Databases

The Yale Face Database [11] (size 6.4MB) contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, and sleepy, surprised, and wink. The collection includes 60 non-facial standard test images (such as cameraman, coins...). These images are considered as negative images for the system, because they do not contain faces.

B. Face detection

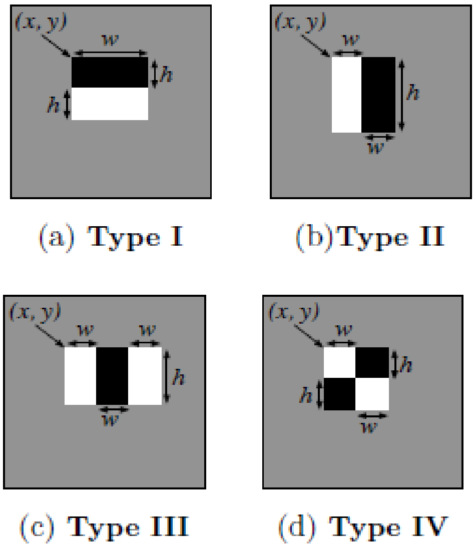
For face detection the Boosted Cascade techniques which use a fast feature evaluation by implementing Haar functions, the AdaBoost method which select a small number of features calculating the haar-likes features. AdaBoost is the effective method for selecting a small number of correct features that does not have a significant variation. For that the learning algorithm selects the features that best classify the positive and negative samples.

In order to calculate these features, the integral image is introduced that is similar to the Summed Area Table see Fig.1.

The classifier was composed through the composition of several weak classifiers through boosting, in which each one is discriminated in function of haar-like feature.

$$g\left(f_i(x); p, \theta\right) = \begin{cases} 1 \;\; if \;\; pf_j(x) < p\theta \\ 0 \qquad otherwise \end{cases} \tag{1}$$

Where $f_j(x)$ is a set of future responses extracted by applying the feature f_j to each training image x_i and associated labels *{y₁,...yₙ}* where y_i ϵ*{0,1}* and a set of non-negative *θ=(p,θ)* and is the threshold value, *p* ϵ{-1,1} and a parity value, ϵ is the value of the error associated with this classifier applied to the training data.



(a) Type I (b)Type II

(c) Type III (d) Type IV

Fig.1. The four types of the features used in Viola & Jones. The figure show how the features are parametrized by the four positive integers *(x, y, w,h)*. Where B(*x,y,w,h*)-B(*x,y+h,w,h*) and B(*x+w,y,w,h*)-B(*x,y,w,h*).

After that, the weighted mean of the positive examples and negative examples are computed.

$$\mu_p = \frac{\sum_{i-1}^{n} \varpi_i f_j(x_i) y_i}{\sum_{i-1}^{n} \varpi_i y_i}, \quad \mu_N = \frac{\sum_{i-1}^{n} \varpi_i f_j(x_i)(1-y_i)}{\sum_{i-1}^{n} \varpi_i (1-y_i)} \qquad (2)$$

Also their error associates with the two possible values of the parity.

$$\in_{-1} = \sum_{i-1}^{n} \omega_i \left| y_i - g\left(f_j(x_i); -1, \theta\right) \right| \qquad (3)$$

$$\in_{1} = \sum_{i-1}^{n} \omega_i \left| y_i - g\left(f_j(x_i); -1, \theta\right) \right| \qquad (4)$$

From where the threshold is set to:

$$\theta = \frac{1}{2(\mu_p + \mu_M)} \qquad (5)$$

This detected the relevant characteristics for classification. In this case 10 characteristics were used. Once the discriminate function was fixed and the decision threshold given by the classifier was tested with the database.

The Yale Face Database and the 60 negative images were tested for face detection algorithm. Also were used the pre-trained haar cascade classifier created by Rainer Lienhart [11], a stump-based 24x24 AdaBoost frontal face detector, included in the OpenCV library.

In order to use OpenCV Haar Feature-based Cascade Classifier, was necessary to solve two problems:

- Integration of multiple detections

The detector is insensitive to small changes in translation and thus multiple detections occur around each face. However the idea is to have just one bounding box detection per face. Therefore, it is useful to prune the detected sub-windows so that overlapping detections are summarized by one detection.

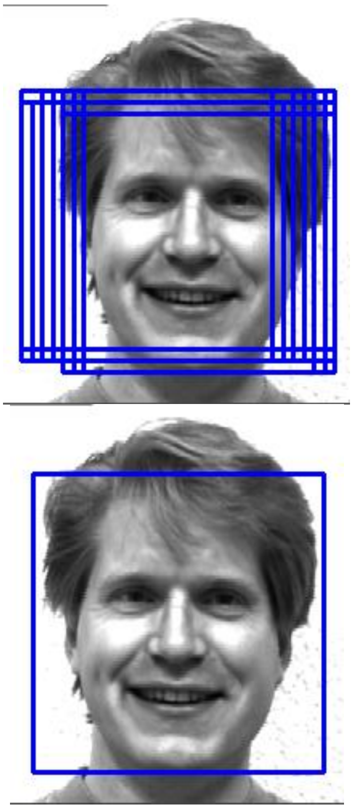


Fig.2. Example of a face detection.

OpenCV function detects MultiScale and receives the parameter minNeighbors, which is the minimum number (n-1) of neighbor rectangles that makes up an object. All the groups of a smaller number of rectangles than minNeighbors-1 are rejected. If min neighbors are 0, the function does not any grouping at all and returns all the detected candidate rectangles.

- Face detection over multiple scales

Since the images are not 24x24 pixels like the classifier windows, the detector needs to be scanned across an image at multiple scales in addition to multiple locations. In order to do that, is necessary to rescale the search window between the subsequent scans. The parameter scaleFactor determines the factor used to rescale, e.g 1.1 means increasing window by 10%.

To determine the values of these parameters to optimize the facial detection system, a test on the database was performed, considering the different values for minNeighbors, and varying the scaleFactor from 1.05 to 1.30 in steps of 0.05. The results are shown in the Fig.3.

As it can be seen, all three configurations perform rather well. Nevertheless, it was choosen minNeighbors=4 because the results are slightly better. For this value, the following accuracy results were obtained for different scale factor, Table 1.

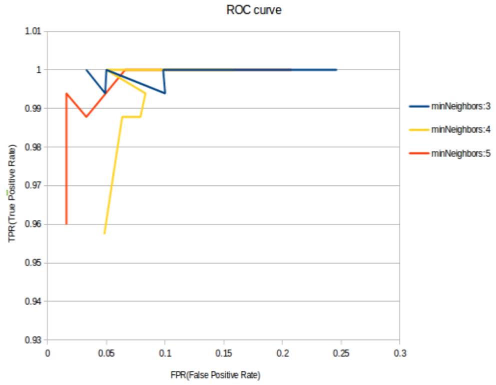


Fig.3. Rescaling of the images.

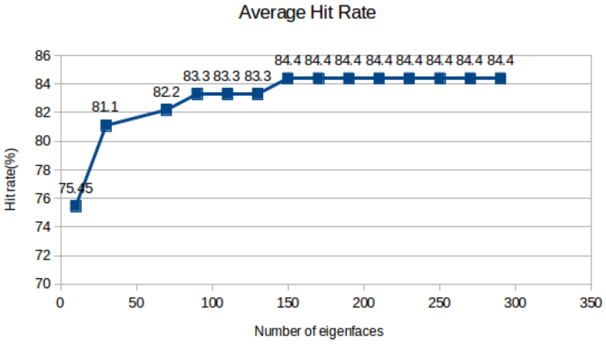


Fig.4. Average Hit Rate for face recognition.

TABLE I. SCALE FACTOR VERSUS ACCURACY

scaleFactor	1.05	1.10	1.15	1.20	1.25	1.30
Accuracy	0.964	0.9822	0.973	0.9822	0.9911	0.9733

From the results the chosen value for the parameter scaleFactor is 1.25 due to its accuracy is the best with 0,9911.

C. Face recognition

With the obtained face detector parameters set now is necessary to evaluate the developed face recognition system, implemented using the **FaceRecognizer** class of OpenCV library [12].

For this experiment, were used 81.81% (9 images of each subject) of the Yale database for training the eigenvector model, and the remaining 18.18% (2 images of each subject) for testing the results. For the testing was choosing the images randomly so it does not influence in the results. The experiment was performed 3 times, changing the training and testing images each time.

After training the model, several tests by varying the number of components (eigenfaces) kept for the PCA were conducted. The figure below shows the average hit rate for the 3 tests. As it can be seen, the system reaches its maximum hit rate with 150 eigenfaces and then stabilizes.
Therefore, the number of components is set to 150, and out final system performs with an 84.4% of success.

IV. Conclussions

The paper deals with the results of the face detection and recognition system for blind people in real time. Two main aspects on machine learning compose the work: the object detection based on boosted cascade and the face recognition based on Eigenfaces. The system is developed on Raspberry hardware. From the results it can conclude that the average hit rate of the recognition system is of 84,4%.

Acknowledgment

The work was supported by the GV/2014/015 project.

References

[1] L.Dunai, G. Peris-Fajarnes, E. Lluna, B. Defez, "Sensory Navigation Device for Blind People," The Journal of Navigation, Vol 66, pp. 349-362, 2013

[2] M. Eimer, "Multisensory Interaction:How Visual Experience Shapes Spatial Perception,"Curent Biology, Vol 14, pp.115-117, February 2004

[3] X. Chen, A. L. Yuille, " A Time-Effect Cascade for Real-Time Object Detection: With applications for the visually impaired," 1st Intern. Workshop on Comp. Vision Applications for the visually CVACVI, June 20, 2005.

[4] P. Viola and M. Jones, "Robust Real-Time Face Detection", I*nternational Journal of Computer Vision*, Vol. 57(2), pp. 137–154, 2004

[5] D. Chen, J-M. Odobez, H. Bourlard, "Text detection and recognition in images and video frames," Pattern Recognition, Vol. 37, pp. 595-608, 2004

[6] R. Parlouar, F. Dramas, M. J-M Macé, C. Jouffrais, "Assistive Device for the Blind Based on Object Recognition: an Application to identify Currency Bills," 11th International ACM SIGACCESS Conference on Computers and Accessibility ASSETS, pp. 227-228, October 25-28, 2009

[7] A. Guaita, C. Catacta, F. Saenz,, P. Bernal, "Desarrollo e Implementación de un Software de Reconocimiento de Dólares Americanos Dirigido a las Personas con Discapacidad Visual Utilizando Teléfonos Móviles Inteligentes con Sistema Operativo Android,"

[8] M.A. Turk & A. P. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. Vol 3, No. 1. 71-86, 1991

[9] M.A. Turk & A. P. Pentland, "Face Recognition Using Eigenfaces," Computer Vision and Pattern Recognition Conference, 1991. Proceedings CVPR'91., IEEE

[10] Krishna Sh., Little G., Black J., Panchanathan Sh., "A wearable face recognition system for individuals with visual impairments," *Proceedings of ASSETS'05*, October 9–12, 2005, Baltimore, Maryland, USA, pp.106-113

[11] http://vision.ucsd.edu/content/yale-face-database contained in the file haarcascade_frontalface_default.xml