# Biometrics using face recognition

**M. Koupil[1] and J. Odstrčilík[1]**

[1]Department of Biomedical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic

E-mail: xkoupi01@vutbr.cz, odstrcilik@vut.cz

**Abstract**—This paper is focused on a face detection in a picture and subsequent recognition of the face in its respective database. Face detection had been implemented using Viola-Jones algorithm. To recognize the face afterwards, PCA had been used. Implemented algorithm had been tested on freely accessible biometric databases.

**Keywords**—Biometrics, face localization, Viola-Jones algorithm, face recognition, PCA

## 1. INTRODUCTION

Among biometric systems, ID systems, that are based on working with face features, dominate. Compared to other metrics, they are significantly more popular for various reasons: Human faces are generally unique and universal [1], also willingness of people to provide a picture of themselves is higher compared to other metrics. It finds its use in criminalistics (e.g. database of fugitives) and it is getting more common in commercial applications (e.g. authorization in mobile devices).

Biometrics of face generally consists of two parts: Localization - finding the face within a picture, and Recognition – its identification in given database. There are many various methods for a face localization and recognition, and even though there is a general scheme, individual methods may vary quite significantly.

The goal of this paper is to implement algorithm performing face biometrics using freely accessible picture databases as input data.

## 2. BIOMETRIC DATABASES

For the testing of implemented algorithm for face localization and recognition, images from free picture databases have been used [2],[3]. Algorithm has been tested on five sets of pictures from these databases. For purpose of this algorithm, only pictures that were taken frontally were used. Individual datasets possess high intra-class variability (people on pictures have different facial expressions, wear different hats, have additional accessory…).

**Figure 1:** An example of training pictures from one of the datasets. High intra-class variability is clearly visible.

## 3. METHODS

### 3.1 Viola-Jones Algorithm

For face localization, Viola-Jones Algorithm has been used [4]. By its nature, it's a method based on

appearance. It uses detectors of characteristic features of the face – Haar filters. The filters scan through the picture in various scales and marks features, that could be part of the face. Classification algorithm *AdaBoost* is used to select the best filters. *AdaBoost* is a linear combination of a group of weak classifiers (=simple Haar filters) which are iteratively turned into strong classifier. To avoid applying all the filters on a window, cascade of classifiers is presented. Several filters are put together in different stages of classifiers and scan sub-windows. If a sub-window fails to include part of the face, it is discarded. Sub-window which will pass all stages is considered to be a part of the face.

For implementation of Viola-Jones Algorithm, library *openCV* has been used. For face detection in a picture, pre-learned model [5] was used.

### 3.2. PCA method

For face recognition, principal component analysis [6] has been used. It is a holistic method, meaning it perceives the face as a whole. It converts the dataset to the new space called *eigenspace*, where individual pictures would be ordered by variance from the mean picture. The mean picture is calculated via equation:

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} Xi, \tag{1}$$

where $\overline{x}$ denotes mean picture, which is counted as a sum of input vectors divided by the number of them.

For training, libraries for machine learning *scikit-learn* [7] was used. From the order of pictures in the new space, we can see that the pictures are ordered by importance for given dataset.

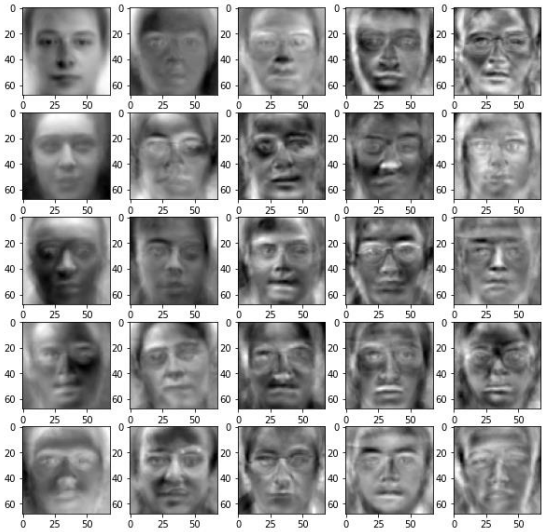

**Figure 2:** The most important 25 pictures in eigenspace of one of the datasets.

In terms of dimensionality reduction, it is possible to heuristically determine number of most important pictures for further calculations. If we were to perform reconstruction of any original image, we would lose minor details, with the price being reducing dimensionality, therefore reducing computational expense. To assign input testing image to the training database (=to recognize it), we need to calculate vector of weights for each training image. Weight vector of input testing image will be compared to weight vectors of every individual image of the training database. The most similar one will be declared as a match.

## 4. RESULTS

As mentioned before, the algorithm has been tested on five different sets of pictures. In Dataset 1, there

are 27 white people, each one has 10 training pictures. In dataset 2, there are 10 Asian people, each one has 13 training pictures. In datasets 3,4 and 5 there are 20 people of various races, each one has 15 training pictures. Intra-class variability in the last 3 databases is higher than in the first 2. While in the first 2 databases hairstyles and facial expressions changed, in the latter 3 also additional accessories, such as glasses or hats, appeared.

Given the fact, that face localization, *frontal face* Haar filters have been used, some of the detections have not been performed. One of the reasons of a failed localization was that part of the face was missing on the picture, another reason might be capturing the face from inconvenient angle.

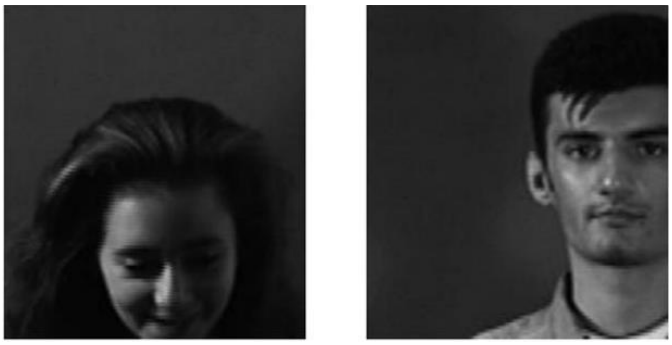

**Figure 3:** Examples of pictures inconvenient for localization

| *Dataset* | *Rate of successful localization* | *Percentual rate* |
|---|---|---|
| *Dataset 1* | *270/270* | *100%* |
| *Dataset 2* | *129/130* | *99,2%* |
| *Dataset 3* | *300/300* | *100%* |
| *Dataset 4* | *297/300* | *99%* |
| *Dataset 5* | *298/300* | *99,3%* |

**Chart 1:** Rate of successful face localization.

Pictures, at which the localization was not performed correctly, were omitted from the database. Because it was a very small number of pictures, it did not interfere with next steps of the algorithm.

After localization of a face on each picture of the datasets, PCA was performed so the testing pictures can be assigned to their respective datasets. Recognition success of testing pictures in terms of their training datasets is stated in the chart:

| *Dataset* | *Rate of successful recognition* | *Percentual rate* |
|---|---|---|
| *Dataset 1* | *20/27* | *74%* |
| *Dataset 2* | *9/10* | *90%* |
| *Dataset 3* | *20/20* | *100%* |
| *Dataset 4* | *18/20* | *90%* |
| *Dataset 5* | *20/20* | *100%* |

**Chart 2**: Rate of successful face recognition.

On following picture there is an example of successful recognition:

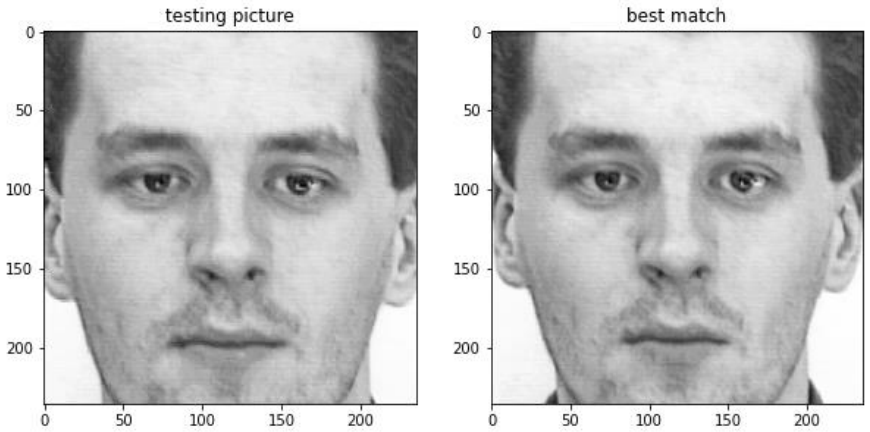


**Figure 4:** An example of successful face recognition

## 5. CONCLUSION

This paper is focused on biometrics using face recognition. In practice, the biometrics was performed in two steps: face localization and face recognition. Viola-Jones Algorithm was used for face localization and PCA was used for face recognition.

Algorithm was tested on five different picture datasets. At face localization, average success rate was 99,5 % and at face recognition, average success rate was 90,8 %.

Lowest success rate of face recognition was registered when Dataset 1 was tested. There might be more reasons why. One reason could be that there weren't enough training images for to algorithm to learn properly. Another possible reason is that because the individuals shared the same ethnicity, there were more similarities between the them, so there is a bigger chance of wrong registration.

Algorithm was also tested on all the databases combined, the PCA method was then applied to pictures of all the datasets. In this case, success rate was 89,6 %.

## REFERENCES

[1] DRAHANSKÝ, Martin a Filip ORSÁG. Biometrie. Brno, 2011.

[2] Face Databases [online]. [cit. 2021-12-08]. Dostupné z: http://web.mit.edu/emeyers/www/face_databases.html

[3] SPACEK, Libor. Libor Spacek's Facial Images Databases [online]. 2009 [cit. 2021-12-08]. Dostupné z: https://cmp.felk.cvut.cz/~spacelib/faces/

[4] VIOLA, Paul a Michael JONES. Rapid Object Detection using a Boosted Cascade of Simple Features [online]. 2001, 9 [cit. 2021-11-22]. Dostupné z: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

[5] OpenCV: Cascade Classifier [online]. [cit. 2021-12-07]. Dostupné z: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[6] SAXENA, Pawan. Face Recognition Using Eigenfaces (PCA Algorithm) [online]. 2021 [cit. 2021-11-30]. Dostupné z: https://www.geeksforgeeks.org/ml-face-recognition-usingeigenfaces-pca-algorithm/

[7] Pedregosa et al. Scikit learn [online]. 2011 [cit. 2021-12-09]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html