



EVALUATING THE APPROACHES TO THE BASIC FACE RECOGNITION SOFTWARE

SULOVSKA, K[aterina] & NECKAR, P[avel]

Abstract: *The face recognition software is usable in many systems. It can be used as a log on to the system by the web camera running in the background to continuously monitor the user's identity. In case of necessity, the application can shut down or log out automatically. Furthermore, the systems can be used for the long-distance surveillance at the airport, metro stations, and other public areas to capture wanted persons. As no system is perfect, it is important to constantly adapt to new trends and technologies. Before the introduction of new software to the market, it is also necessary to thoroughly test this tool in normal mode. This paper deals with the basic creation of such software and its testing in common conditions.*

Key words: *face recognition biometrical systems, system evaluation, pattern recognition*

1. INTRODUCTION

The person is able to recognize other people by various characteristics. One of the most remarkable is the face. Without thinking, the brain automatically compares the face of the person standing in front of us with the image in our memory. During the identification, the knowledge from the anthropology is used. If we measure the anthropologically significant points (minutiae), which match with data from other photography, then we can designate the faces as identical. There are a lot of methods for locating these points in the image. A lot of them use a binary image formed by thresholding, whereon the edges and corners detectors are applied.

In the machine face cognition a lower quality image or multi-colored background must be taken into account. In such a case a number of different filters to get rid of image noise, highlight the contours or adjust brightness are used. If we use the correct method for the wrong image, the necessary points can be found although the image was completely unusable before this process. The human face itself is not always completely the same. Apart from the effect of the time on the face, the ambient temperature can also change the appearance of the face - mainly the skin color. This fact can influence the results of the system based on the color properties of the face.

This paper aims to introduce the simplest form of the face recognition software using the available open libraries and basic approaches for the localization and the recognition of the face.

2. EQUIPMENT

The software was created in the development environment Visual Studio 2010 by the Microsoft. The program uses the .NET framework, and the source code was written in the C#. While creating the software, some functions of external libraries of the Emgu CV (ver. 2.2.1) and the AForge.NET (ver. 2.1.5), which are both available as open source licenses, were used. The software was tested on three computers, the desktop computer and two laptops. The computer equipment has following characteristics:

- Intel Pentium 2.6 GHz, 1.5 GB RAM, Win XP,
- Dual-core processor 2.2 GHz, 3 GB RAM, Win 7,
- Intel Atom N455 1.66 GHz, 1 GB RAM, Win 7.

3. METHODS

3.1 Anthropological face identification

The anthropological knowledge is one of the resources for the utilization in the biometric systems technology together with other scientific disciplines. For the unambiguous face identification, 12 points located on the face (area of eyes, mouth, nose and ears) are necessary to be determined:

- inner and outer corners of the eyes,
- horizontal corners of the lips,
- the nose tip,
- transition of the nose to the forehead,
- connection of the earlobe and the face,
- points on the ear cartilage protecting the alvearium.

Then are those points connected with vectors and their length is measured. The face is recognized according to the length ration of these vectors. It is evident that the face on the photograph can be rotated in three axes, which change distances of the points.

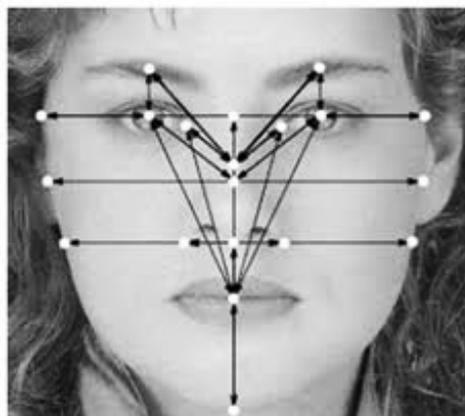


Fig. 1. Example of the points array

3.2 Finding the face, eyes, mouth and nose in the image

For the right functionality of the software, it is important to maintain the proper exposure of the face. The image cannot be scanned in the rooms with low intensity of the light, but it may not be overexposed too.

The Emgu CV library offers the haarcascade files for the localization of the face and its parts. First, the face must be found in the image with the file haarcascade_frontal-face_alt.xml. The DetectHaarCascade method is then used on the image transformed to the gray scale to return the area with required objects. This method can also localize edges by the Canny detector. After this process, we get the square or the rectangular area with the face, where the borders are depicted

by the method Draw. The parameters of this method are: object to be depicted, color of border lines of the object and their strength. An occasion when we get the right eye together with the localization of the left eye (or even with the mouth) may occur. This issue can be solved by the vertical and horizontal geometrical division of the area with the face. Then the gray scale image is setup to this size.

Finding the corners of the eyes and the mouth, and the center of the nose is very important in order to identify the person correctly. To locate the corners of the eyes and the mouth, the GoodFeaturesToTrack method from the Emgu CV library is available. This function is able to find the array of points at right setup and quality of the picture. The point with the lowest and the highest value on the x-axis is chosen from this array. The parameter of the minDistance function is dependent on the size of the localized face. For finding the center of the nose, the previous square is used, respectively the center of this square.

```

1 Image<Bgr, byte> m_OriginalImage = new Image<Bgr,
  byte>(bitmap);
2 Image<Gray, byte> m_GrayScaleImage =
3     m_OriginalImage.Convert<Gray, byte>();
4
5 Rectangle faceRectangle = new Rectangle();
6 string m_FileHaarFace = "Way to file";
7 HaarCascade faceHaarObj = new
  HaarCascade(haarCascadeFile);
8 MCvAvgComp[][] facesDetected =
9     grayScaleImage.DetectHaarCascade(faceHaarObj,
10     1.1, 10,
11     HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
12     new Size(20, 20));
13 foreach (MCvAvgComp face in facesDetected[0])
14 {
15     faceRectangle = face.rect;
16 }
17 m_OriginalImage.Draw(faceRectangle, new
  Bar(Color.Red). 3);

```

Fig. 2. Obtaining the right side of the face (Košutek, 2011)

3.3 Interpretation of the found data

Initially, it is essential to count the distances of single points. This can be done by an easy function based on the calculation of the point in 2D space, where the position of each point is done by the x- and y-coordinates by the formula:

$$h = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (1)$$

As the size of the face in the image may not be the same as the size in the image during registration, the ratio of distance of each point in the registered and submitted image is calculated. From these values, the average is recognized, and if the difference between measured values and the average is lower than the chosen toleration, the distance is marked as identical with the registered distance. We can also make histograms of both images:

$$n = \sum_{i=1}^k m_i \quad (2)$$

4. RESULTS AND DISCUSSION

As the software is very simple, it is necessary to install the .NET Framework 4.0 before launching the own application.

The training files haarcascade are placed into the folder HaarCascadeXML. As testing shows, the requirement for correct operation of the software is the size of the picture c. 700×900 pixels with minimum resolution 70 ppi. The width of localized head should be from 400 to 600 pixels. The problems can occur when the improper light makes a shadow in the place of the corner. The software is also resistless to the turning of the head, which should therefore be minimal, and the face should not be covered (hats, scarf, etc.). To secure the data of the user in the database, the 256 bit Rijndael was implemented.

The testing was done for four users in suitable light conditions. The toleration was set-up to 60 %. In 6 out of 100 attempts was the person recognized, which means that the software works correctly for 94%. After setting-up the toleration to 80 %, the number of correct attempt decreases to 83 %. However, the simplicity of the program enables testing only in laboratory-like conditions.

5. CONCLUSION

Nowadays, the top research groups are creating highly sophisticated softwares for the face recognition. Some of these softwares are almost perfect so that creating something totally different based on a novel approach is the matter of coincidence of few factors.

Machine face recognition is a young science discipline constantly evolving. The aim is to minimize the influence of inferior image quality, the effect of irregular illumination of the face, and the head rotation. The software solution introduced in this paper is able to work properly only in suitable conditions. This state is not ideal at all, and the software should be improved by new algorithms and approaches in order to be competitive with modern solutions. This will lead to completely new application created in C++ in order to fasten the processes.

6. ACKNOWLEDGEMENTS

This paper is supported by the Internal Grant Agency at TBU in Zlín, project No. IGA/43/FAI/10/D (Research on biometrical systems in term of their endurance, reliability, credibility and integrity), and by the European Regional Development Fund, project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089. We would also like to thank Mr. Pavel Košutek for his help on this project.

7. REFERENCES

- ABdallah S. (2007). *Investigation of new techniques for face detection*, Blacksburg, Virginia, 2007, Virginia polytechnic institute
- Dobeš, M. (2008). *Image processing and algorithms in C#, BEN*, Prague 2008, ISBN 978-80-7300-233-6
- Kaněčka, P (2007). *Locating the significant points in raster image*, Diploma thesis, Brno, 2007, Brno University of Technology
- Košutek, P. (2011). *Software-based access control system based on facial biometrics*, Diploma thesis, Tomas Bata University in Zlín, Zlín 2011
- Peng, K. et al. (2005) A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles. *Computer Science and Technologies*, 2005, 3
- Rak, R. et al. (2008). *Biometry and identity of the person in forensic and commercial application*, Prague, Grada Publishing, 2008, ISBN 978-80-247-2365-5
- Bauldauf, K. (2011) *Face recognition threatens privacy*. Online, cited 2011/08/14, Available from: <http://www.teachtech.biz/2011/08/face-recognition-threatens-privacy/>