FACIAL RECOGNITION SYSTEMS

A. M. Hamrayev, P. S. Myradov, M. M. Hojamammedov

The State Energy Institute of Turkmenistan

On the basis of this program, the program that registers employees by face at the entry-exit points is designed for the security of electronic document exchange and the use of cryptographic methods of data protection. The program allows solving the following problems in data transmission systems and computerized systems, i.e., the reliability of information exchange in electronic document exchange and security, high level of confidentiality of the transmitted information, resistance to various crypto-attacks, it allows to carry out tasks such as simultaneous implementation of data encryption and decryption. This program can be installed and used in any office or business.

Facial recognition systems. The process of face recognition is usually a set of different tasks that serve to recognize a person from a digital image or video clip. In general, the process looks like this: after the system receives an image from the camera, the face boundaries are determined by algorithms (the face extraction phase). After that, the recognition phase begins, where the face is modified (brightness, alignment, resizing, etc.) and brought to a certain shape. The features are then calculated and directly compared to benchmarks stored in the database. This final stage of comparison is called identification or verification, depending on the system.

Verification: Comparison of samples in a "1:1" scheme. To identify an individual, the system compares the biometric sample with a biometric template stored in the database and asks, "Is this person the same person as the template?" answers the question. Identification: Comparison of samples in a "1:N" scheme. To determine the identity, the system compares the biometric sample with all facial templates stored in the database and asks "who is this?" answers the question.

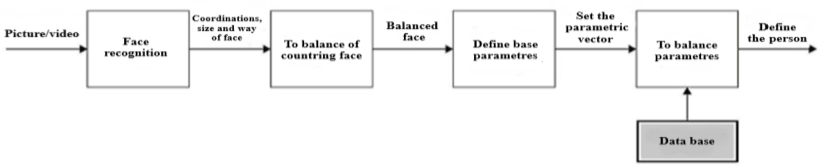

Figure 1. General algorythms of defining face

Recognise face by person. Humans can identify their surroundings in tens of milliseconds. Such a high rate of object recognition is possible because our brains make constant predictions about what we see and compare these predictions with information from the outside world. There are three main steps in human face recognition:

1) to determine the physical characteristics of the subject;

2) identify the person, based on which we determine whether the person is known to us or not;

3) we know the person, but we still don't know whether we know his name or not.

Scientists have found that certain areas of the human brain are activated at each stage. As psychologists point out, face recognition is more related to the cognitive side of emotion. In fact, face recognition occurs as follows: the brain constantly compares what it sees with what it holds in long-term memory. Ironically, this is how almost all face recognition algorithms already embedded in the database work. For example, when we look at an object such as a watch, we compare what we see with the features specific to the mental image of the watch. Not all watches are the same, and although some models of this object may differ from the prototype in mind, each watch has key features that are unique to it, such as the minute and hour hands that aid recognition, and the dial. The image of the object is then classified into a specific category and stored in memory. The more hours our brain stores, the easier it is to recognize something new.

Object classification is usually the final step in the recognition process, but in the case of face recognition, it's just the beginning. If it is enough to recognize a clock as a clock, it is not enough to recognize a human face as a human face. Almost immediately we judge a person's gender and age, race, and even whether we like them or not. In addition, we immediately determine whether this face is familiar to us. If a person is familiar to us, we immediately start receiving information from them, just like a face recognition algorithm. It can identify a specific person and then access the information and send a signal to the crime agency depending on why it was created.

Machine authentication. The problem of face recognition and identification is one of the first practical problems that motivated the creation and development of the theory of object recognition and identification. There are nine categories that correspond to Gnostic places and create visual images:

1) objects that can be manipulated;
2) partially controllable objects;
3) uncontrollable objects;
4) faces;
5) facial expression;
6) living beings;
7) printed characters;
8) manuscript images;
9) characteristics and location of light sources.

The problem of face recognition was considered in the early days of computing. Certain companies have been actively developing automated human identification systems for more than 40 years, and today: Smith and Wesson (ASID system - Automated Suspect Identification System); ImageWare (FaceID system); Imagis, Epic Solutions, Spillman, Miros (Trueface System); Vissage Technology (Vissage Gallery System); Visionics (FaceIt System).

Different methods have been proposed to solve the problem of face recognition, among them, based on neural networks, based on Karhunen-Loeve partition, algebraic moments, lines of equal intensity, and elastic (deformation) comparison standards. The development of recognition algorithms focuses on the automatic selection of facial elements (eyes, nose, mouth, chin, etc.) from various images: face, profile and arbitrary angle. In addition, these geometric features are used to solve recognition problems. A common feature in describing these approaches is the lack of comparison in a statistically significant database.

Geometric comparison based on the definition of facial elements – facial elements: eyes, nose, mouth, chin, etc. A face can be recognized even though individual facial elements are not visible enough. The idea of scaling is to find the relative position and features of the individual elements of the face. Even when face elements are entered manually, the computer shows very good results.

A benchmark comparison is a built-in image representation as a byte array – the intensity magnitudes are compared to the large surface – a metric that matches the benchmark. There are several ways to prepare and display benchmarks. Several benchmarks are used for recognition from different perspectives.

The reference comparison scheme in Bruce V.'s work is modified enough to be called correlation-extremal. It uses image normalization that translates to a gradient size map and has no edge maps. One of the successful findings is the use of multi-dimensional and small-scale references for the eyes, mouth and nose. Elemental element identifiers are based on these approaches. It should be noted that the next step is constructive: first detect the eyes (by comparison), then automatically adjust the image for scale and orientation.

It can be seen that this approach has elements of recognition based on whole face reference: YE (eyes) is used for image normalization and reference comparison is done separately for individual face features (eyes, nose, mouth). However, experiments show that face recognition is emerging in an architecture that combines a face recognition approach with an element matching approach.

A recognition scheme in neural networks is of interest. In particular, the use of a network of hyperbasic functions in feature vector synthesis for arbitrary angle recognition of 3D objects. In this case, the mesh inputs include the surface elements, including their position in the image. The hyperbasic functional grid, the amplitudes of the gradients for each pixel and the centers of the corresponding standards, different centers in different schemes, is similar to the scheme described earlier for comparing Face Elements templates. In these correlation coefficients, instead of the maximum method, linear classification by Gaussian functions of the correlation coefficients can be fitted. The problem of the dependence of the recognition results on the shooting angle can be solved in several ways. If there are images taken from different angles for each individual, the same recognition schemes can be used at the expense of increasing the computational cost. Using hyperbasis functions - classifications with the possibility of interpolation between different projection points is very risky. However, in reality there may be only one face image to create a template. Of course, a single image of a 3D object (without shadow) does not contain enough information. However, if the object belongs to a group of similar objects (prototypes) for which different projection points are known, reasonable extrapolation is possible and the correct projection can be presented for this object from the 2D projection alone. Humans are able to recognize faces rotated 20-300 from the frontal projection. Perhaps they are using information about the structure of the normal face. Another solution to this problem is to use 3D face models to support recognition in non-face images. As R. Brunelli pointed out, it is possible to identify and solve problems such as working on an expert database related to obtaining other face projections using knowledge about the projections of other typical objects of this class.

For different contours of facial elements, different methods are used to extract them in the original portrait. The eyes and mouth have fixed geometric shapes, so they are extracted according to the deformation reference model. Other facial features such as eyebrows, nose, and facial contours vary so much that an active contour model is used to detect them. Figure 3 shows all the features of the face that are intended to be detected in automatic facial recognition, which are used in portrait testing and ensure the validity of the method. Deformation standard model. The variable parameters are defined by the parameters provided by the a priori knowledge about the expected shape of the UE and are numerically determined during the learning process during contour decoding. The standards are highly flexible and can be numerically compared when their dimensions and other tuning parameters are changed, and the obtained parameter values can be used to describe specific surface elements. Deformation references interact dynamically with the current digital image. The energy function is defined by a set of components that capture the image reference. Surface elements based on characteristics of intensity slice graphs, such as maxima and minima, edges, and intensity values. The minimum energy function corresponds to the best choice for a given image. Usually, deformation standards are used to define eyes and mouths.

Recognise face by using OpenCV library

First, let's learn how to recognize a face in a photo. First, you need to find where the person's face is in the picture and not confuse it with the clock on the wall and the cactus on the window. A simple task for a human may not seem so simple for a computer. To find a face, we need to select the main components such as nose, mouth, eyes, lips. For this we will use the templates shown in Figure 2.

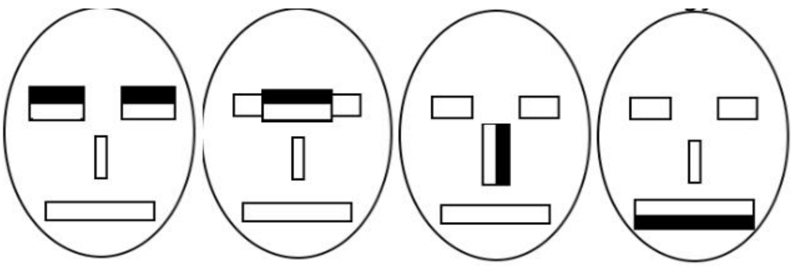

Figure 2. Basic templates

Using of symbols of Haar. If the templates, if they are the first, match certain areas of the image, we assume that the image has a human face. For each of them, the difference between the brightness of the white and black areas is calculated. This value is compared to a standard and decides whether a part of the human face is present or not. This method is called the Viola-Jones method (also known as Haar cascades). Let's imagine that there are not only big faces in the picture, but also small ones. If we apply the templates to the whole image, we won't find the face there because it will be smaller than the templates. A sliding window method is used to search the entire image for faces of different sizes. It is in this window that the initials are counted. The window appears to slide across the entire image. As each image is passed, the window is enlarged to find larger-sized surfaces.

A face has been found in the photo, but it takes a few more steps to recognize a specific person. We will use the Local Binary Patterns algorithm to solve this problem. Its essence is that we divide the image into slices and compare each pixel in each slice with its 8 neighboring pixels. If the value of the central pixel is greater than its neighbor, we will write 0, otherwise we will write 1. So we will get a specific number for each pixel. In addition, based on these numbers, a histogram is calculated for all the segments that we have divided the image into. The histograms from all the slices are combined into a vector that characterizes the image as a whole. If we want to know how similar two faces are, we need to calculate and compare such a vector for each of them, the calculation of the vector is shown in Figure 3.

5	8	1		1	1	0
3	4	6		0		1
7	1	3		1	0	0

Figure 3. Calculation of LBP weight

The vector is written to line 11010001. Figure 4 shows the face detection algorithm. The CascadePath parameter contains the name of the file with values ready to refer to. This file was downloaded from GitHub.

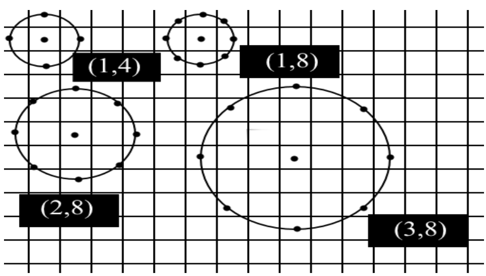

Figure 4. LBP radius

The following parameters (8,8) describe the dimensions of the areas shown in Figure 5, into which we divide the faceted original image. The smaller it is, the more it will be and the better the detection will be.

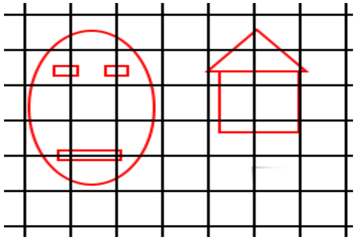

Figure 5. Dividing to districts

Finally, the last value is the confidence threshold parameter, which defines the threshold value for face recognition. The lower the confidence, the more confident the algorithm is that the image shows a face it knows. When the threshold is low, the confidence means that the algorithm considers only this face as unfamiliar. In this case the threshold is 123.

Algorithm description and implementation. First, we need to familiarize ourselves with a simple face recognition algorithm. To recognize a specific person in real time, you need to create a database of their photos. Then translate it into iml format so that the algorithm can compare the frames in the database and the ones coming from the camera. Finally, the algorithm must inform the user who is in front of the camera.

A proper implementation of such a problem would be to divide it into several algorithms that work independently and each performs its own task. Figure 6 shows the real-time face recognition algorithm.

```
import numpy as np
import cv2
faceCascade =
cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cap.set(3,640)
cap.set(4,480)
while True:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.2,
        minNeighbors=5,
        minSize=(20, 20)
    )
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
    cv2.imshow('video',img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

Figure 6. Recognise face

After importing numpy and cv, you need to import an xml file with positive and negative examples of faces. First, the algorithm requires many positive images (images of faces) and negative images (images without faces) to train the classifier. We need to extract functions from it. This step would take too much time to describe, so this section will be omitted below. To run the algorithm multiple times, you need to run the loop with an exit condition.

The img variable accepts a 640X480 resolution image that the algorithm will run on. In black and white format, the program is easier to recognize the face, because the algorithm working on Haar features can detect sudden changes in shadows with greater accuracy, gray is still the same img image, but black and white.

By creating a classifier function, we can not only define the parameters for face detection, but also simplify the task for the algorithm.

So we can create an enlarged pyramid. For example, parameter 1.2 shows that the image will be reduced by 80%, but not once. An Expandable Pyramid is a collection of the same shape, but in different sizes.

minNeighbors - minimum number of neighbors. The image describes which neighbors are in question. The higher the parameter, the more accurately the algorithm recognizes faces.

minSize is the minimum image size that can be mistaken for a face, the smaller the minSize, the better the algorithm will perform.

maxSize - Similar to minSize, but works in reverse, it is not used in the algorithm, so the face found can be any size.

For has nothing to do with face recognition, it just draws a rectangle around the detected face.

Create an xml file. Working with an extended cascade of weak classifiers involves two main steps: a training phase and a detection phase.

Sources

1. Громов Ю.Ю., Дидрих И.В., Иванова О.Г., Ивановский М.А., Однолько В.Г. Информационные технологии. Тамбов. 2015.

2. Абрамян М. Э. Электронный справочник по программированию Версия 4.5 Ростов на Дону, 2005.

3. Абрамов С. А., Гнездилова Г. Ф., Капустина Е. Н., Селюг М. И. Задачи по программированию Вологда, 2000.

4. Шень А. Программирование. Теоремы и задачи. М., 2004.

5. Никита Культин “Основы программирования в Delphi XE” Санкт-Петербург, 2011.