International Journal of Research in Engineering, Science and Management
Volume-1, Issue-12, December-2018
www.ijresm.com | ISSN (Online): 2581-5792

Face Detection and Recognition System Using Raspberry Pi

Ashish Ernest Rahul
Software Engineer, Department of Embedded Technology, Vellore Institute of Technology, Vellore, India

***Abstract*: Face recognition embedded system can be used as an interfacing medium between the computer and human using different faces of a person in order to control the authentication of a computer. In this proposed system, a real time vision based interaction prototype which depends upon face recognition algorithms is designed. An embedded face recognition system based on the Raspberry Pi single-board computer is proposed in this paper. Face recognition system consists of face detection and face localization using Haar feature-based cascade classifier. Face features are extracted using weighted Local Binary Pattern algorithm. Developed system performs one full face analysis in 5 seconds. Comparison of two bio-metric samples is performed in 2 ms. Face recognition is an exciting field of computer vision with many possible applications to hardware and devices using embedded platforms like the Raspberry Pi and open source computer vision libraries like OpenCV.**

***Keywords*: face recognition**

1. Introduction

Traditional ways for personal identification depend on external things such as keys, passwords, etc. But such things may be lost or forgotten. One possible way to solve these problems is through biometrics, for every person has his special biometric features definitely.

Biometrics identification has gained increasing attention from the whole world. Biometrics features that can be used for identification include fingerprints, palm prints, handwriting, vein pattern, facial characteristics, face, and some other methods such as voice pattern, etc. Compared with other biometric methods, the face recognition has the following advantages: The face image acquisition requires no physical contact, so face identification system is non-invasiveness. Since, the face is created in a nearly random morphogenetic process during the gestation, it has little probability to find two people in the world whose face textures are identical. So face recognition is the most accurate method and has the lowest false recognition rate. The face recognition has more stability than other biometric identification methods because the face has much more features than other biometrics and it won't change in people's life. With the advantages of non- invasiveness, uniqueness, stability and low false recognition rate, face recognition has been researched widely and has a broad usage, such as security, attendance, etc. Most of the recognition systems are based on PC. However, the portability of PC is limited by its weight, size and the high power consumption.

Thus results in that the using of face recognition is confined in few fields, and it is inconvenient to use. The way to get rid of the limit of PC is using embedded system. The designed EICSRS platform acquires the images and stores them into the real time database, which in turn later used for comparing the faces of the users to provide access to them or to deny the access to a place or to operate a device. Recent technological advances are enabling a new generation of smart cameras that represent a quantum leap in sophistication. While today's digital cameras capture images, smart cameras capture high-level descriptions of the scene and analyze what they see. These devices could support a wide variety of applications including human and animal detection, surveillance, motion analysis, and facial identification.

2. System hardware design

The whole system is composed by following parts: an image capturing camera, Raspberry Pi board to run image recognition programs on it. DVI compatible monitors also connected with this system during initial stages to preview the captured images and give the user indication. The system block diagram is shown below.

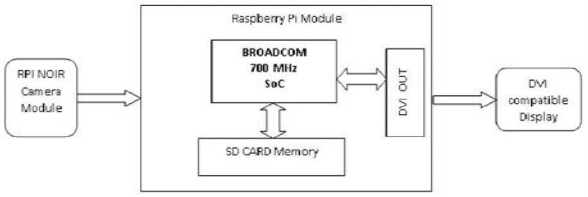

Fig. 1. System block diagram

A. Raspberry Pi board

This board is the central module of the whole embedded image capturing and processing system as given in figure 2. Its main parts include: main processing chip, memory, power supply HDMI Out, Ethernet port, USB ports and abundant global interfaces.

1) Main processing chip

The main signal processing chip used in our system is a Broadcom 700MHz Chip in which CPU core is a 32-bit ARM1176JZF-S RISC processor designed by Advanced RISC Machines, Ltd. It has very rich peripheral. This main processing chip connects a camera and display units.

2) Interfaces

Plenty of interfaces are contained on the Raspberry Pi board, including 2 USB ports through which a Keyboard and mouse can be connected, a HDMI out for connecting HD TVs and monitors with HDMI input or HDMI to DVI lead for monitors with DVI input. Other peripherals like a standard RCA composite video lead to connect to analogue display if HDMI output is not used. Ethernet port is used for networking even though it is optional, although it makes updating and getting new software for Raspberry Pi board much easier. An Audio lead is also provided for getting the stereo audio if HDMI is not used, otherwise HDMI will get digital audio with it.

3) Camera interface

The camera module used in this project is RPI NOIR CAMERA BOARD i.e. Raspberry Pi No IR camera board as shown in the Figure 3. The camera plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver clear 5MP resolution image, or 1080p HD video recording at 30fps. The module attaches to Raspberry Pi, by way of a 15 pin Ribbon Cable, to the dedicated 15 pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor. This camera board which has no infrared filter making it perfect for taking infrared photographs or photographing objects in low light (twilight) conditions. Other features of this camera board are Automatic image control functions, Programmable controls for frame rate 32 bytes of embedded one time programmable (OTP) memory and Digital video port (DVP).

Fig. 2. Pi camera

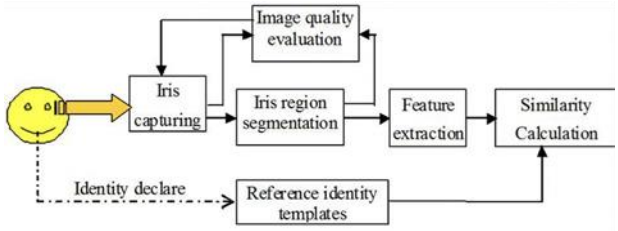

Fig. 3. Block diagram of approach

3. Face region segmentation

To recognize real objects like, in our case, people and their features, a method known as " Haar feature cascade " or Viola-Jones method is supplied in SimpleCV, and in OpenCV as well. The method, in fact, was proposed in 2001 by Paul Viola and Michael Jones in their article "Rapid Object Detection using a Boosted Cascade of Simple Feature", which actually means that it is possible to rapidly identify objects by means of a cascade of consecutive combinations of simple features. The method is a combination of four key components:

1) Comparison characteristics, materializing from rectangular pixel matrices, and known as Haar features;
2) Integral image calculation, starting from the image to process, so to speed up the features detection;
3) The application of the learning method, as for AdaBoost Computer Vision systems;
4) A cascade classifier, so to speed up the detection process.

The algorithm to extract the features comes from the Haar wavelet simplifying up to the point of being ridiculous, the wavelet is represented by two square matrices, one representing the upper part of the wavelet, the other one representing the lower part.

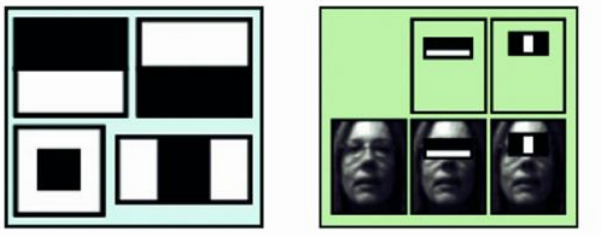

Fig. 4. Face region segmentation

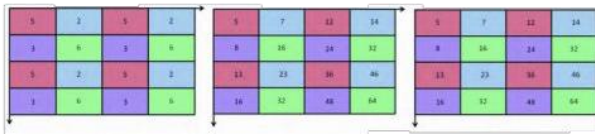

Fig. 5. Face region segmentation matrix

In the model employed for the extraction of the features from the images, the reference matrices have different shapes, such as the ones that can be seen in figure, that are more suitable for determining the shapes belonging to the human body, like the eyes or the nose. From this comes their denomination of Haar Features, to distinguish them from their original meaning. The same picture shows the shape of the features used by OpenCV and SimpleCV. The presence or not of a Haar "feature" in a portion of the picture happens by subtracting the median pixel value that are present in the black "mask" portion, from the median value of the pixels that are present in the clear part of the "mask". If the difference is above a certain threshold value, the feature is considered as present. The threshold value is determined, for each feature, during the function training, to detect particular objects or parts of the human body. The learning process materializes itself when "presenting" to the Vision System the highest possible number of images concerning the "objects " family that we want to identify, and the highest possible number of images that have nothing to

share with the object itself. From the amount of data that are "studied", the threshold values are calculated, for each of the features that, in the case of OpenCV and SimpleCV, are memorized as a file in .xml format.

The portions of the picture that are analyzed are usually formed by 24×24 pixel matrices that have to be "compared" with all the expected features. Here a first problem arises. By processing a 24×24 pixels matrix.

The process consists in assigning to a certain pixel with a certain position within the matrix (image) the sum of all the pixels that are there in the area above and on the left of its position. Starting from the pixel up and on the left and continuing to the right and downwards, the incremental calculation process of the value for each pixel is definitely efficient.

Despite the simplification, the processing volume is still too big to be efficient. The authors of the method noticed that certain features were more significant than other ones, for the purpose of recognizing certain parts, e. g.: the eyes, but were not at all significant to detect, e. g.: the cheeks. Some other features are not at all significant. It can be done by means of a mathematical algorithm for "machine learning", created to optimize the performances of other learning algorithms.

In extremely plain terms, that would horrify purists, its purpose is to research the smallest possible set that would grant a given percentage (for example, 75%) of accuracy as the result for detecting or discarding the required object. For the sake of completeness and greater accuracy, the implementations of the method employ about 6.000 features from the 160.000 initial ones. If we had stopped here, for each 24×24 pixels matrix we should compare 6.000 features. Still definitely too long.

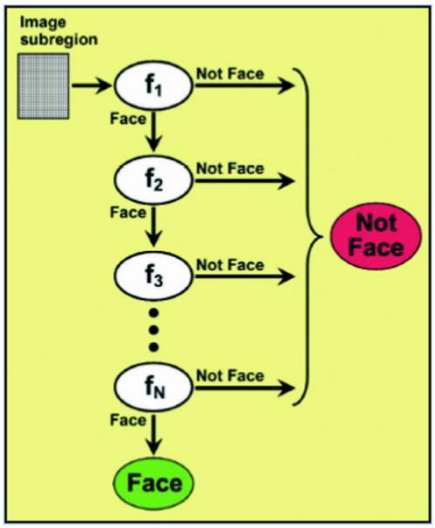

Fig. 6. Training algorithm for faces detection

4. Eigen faces

The problem with the image representation we are given is its high dimensionality. Two dimensional pXq grayscale images span a $m = pq$ dimensional vector space, so an image with $100X100$ pixels lies in a $10{,}000$dimensional image space already. The question is: Are all dimensions equally useful for us? We can only make a decision if there's any variance in data, so what we are looking for are the components that account for most of the information. The Principal Component Analysis (PCA) was independently proposed by Karl Pearson (1901) and Harold Hotelling (1933) to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is, that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components.

A. Algorithm eigen faces

Let $X = [x1, x2, \ldots\ldots, xn]$ be a random vector with observations $Xi \in R^{d.}$

Compare with u

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Compute the the Covariance Matrix S

$$S = 1/n \sum_{i=1}^{n} (Xi - u)(Xi - u)^{Ti}$$

Compute the eigenvalues and eigenvectors of λ_i and V_i of S.

$$Sv_i = \lambda_i v_i,\ i = 1, 2, \ldots\ldots, n.$$

- Order the eigenvectors descending by their eigenvalue. The principal components are the eigenvectors corresponding to the largest eigenvalues.

The k principal component observed vector $x$are then given by

$$y = W^T(x - u)$$

Where, $W = (v_1, v_2, \ldots, v_k)$

The reconstruction from the PCA basis is given by:

$$x = Wy + u$$

Where, $W = (v_1, v_2, \ldots, v_k)$

The Eigenfaces method then performs face recognition by:

- Projecting all training samples into the PCA subspace.
- Projecting the query image into the PCA subspace.
- Finding the nearest neighbour between the projected training images and the projected query image.

Still there's one problem left to solve.

Imagine we are given 400 images sized $100X100$ pixel. The Principal Component Analysis solves the covariance matrix $S = XX^T$, where size(X)=10000X400 in our example. You would end up with a 10000X10000 matrix, roughly 0.8GB. Solving this problem isn't feasible so we'll need to apply a trick. From your linear algebra lessons you know that a MxN matrix with M>N can only have N-1 non-zero eigenvalues. So it's possible to take the eigenvalue decomposition $S = X^TX$ of NxN size instead:

$$X^TXv_i = \lambda i(Xvi)$$

and get the original eigenvectors of $S = X^TX$ with a left multiplication of the data matrix:

$$XX^T(Xvi) = \lambda i(Xvi)$$

The resulting eigenvectors are orthogonal, to get orthonormal eigenvectors they need to be normalized to unit length.

5. Results

This example shows how to automatically detect a face using feature points. The approach in this example keeps track of the face even when the person tilts his or her head, or moves toward or away from the camera. First, you must detect the face. Use the vision cascade object detector System object to detect the location of a face in a video frame. The cascade object detector uses the Viola-Jones detection algorithm and a trained classification model for detection. By default, the detector is configured to detect faces, but it can be used to detect other types of objects.

Fig. 7. Face detected

The Raspberry Pi is provided with ample of training data and the Pi runs the Eigenface Algorithm on the given data and extracts some features of the face that it find important for recognition and clubs all the features together to form a summarized image of the face of sample individual.

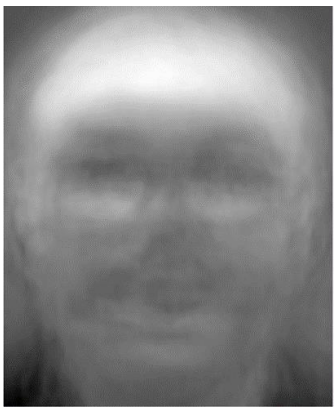

Fig. 8. Summarized image of all the features of my face

6. Future applications

To date, exploitation of smart camera technology has been mainly for industrial vision systems, but a crossover is just starting to take place. Camera technology will begin to enter new applications, for example, in the security and access control markets, in the automotive industry, for collision avoidance, and even – one day – for the toy industry. Even our automobiles may soon be outfitted with miniature eyes. Built into a cruise control system, for instance, such a camera would suddenly alert the driver if it noted a rapidly decelerating vehicle. The cameras could also take the place of the rear view and side-view mirrors, thereby eliminating dangerous blind spots and - in the event of an accident – recording the seconds prior to a collision. Another example would be with intelligent lifts. An office block, with many lifts and floors, may see a lot of people travelling up and down between floors, particularly at high traffic times such as early morning or end of the working day. At the moment, lifts are called by somebody pressing a button and putting in a request for the lift to stop at a particular floor. Connected with smart camera technology, lifts could be routed on demand, working intelligently, stopping only when there was a pre-set number of passengers waiting at a floor – and missing out a floor if too many people were waiting to meet the maximum capacity of the lift.

7. Conclusion

It's a progress of realizing embedded image capturing system. We describe our design method in this paper. Based on these methods, we design the experimental prototype of the embedded image capturing system with Raspberry Pi system. This system is smaller, lighter and with lower power consumption, so it is more convenient than the PC-based face recognition system. Because of the open source code, it is freer to do software development on Linux. Experimental results show that it's an effective method of using Raspberry Pi board to actualize embedded image capturing system.

References

[1] Geraldine Shirley N and S.Jayanthy, "Virtual Control Hand Gesture Recognition System Using Raspberry Pi", ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 7, April 2015.
[2] G. Senthilkumar, K. Gopalakrishnan and V. Sathish Kumar "Embedded Image Capturing System Using Raspberry Pi System", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 3, no. 2, 2014.
[3] Swathi .V and Steven Fernandes "Raspberry Pi Based Human Face Detection", IJARCCE, vol. 4, no. 9, April 2016.
[4] Pallavi V. Hajari and Ashwini G. Andurkar "Review Paper On System For Voice And Facial Recognition Using Raspberry Pi", International Journal of Advanced Research in Computer and Communication Engineering vol. 4, no. 4, April 2015.