

# Interfacing of MATLAB with Arduino for Face Recognition

<sup>1</sup>Kiran R, <sup>2</sup>M Lohith, <sup>3</sup>Yogesh E, <sup>4</sup>Sai Kumar A, <sup>5</sup>Anitha J

## Abstract

This paper explains proposed algorithm for face recognition using image processing and manipulation of the output pin state of Arduino board with ATmega328P controller by tracking the face of a human. The face recognition algorithm has been developed on MATLAB platform by the combination of several image processing algorithms. Using the theory of Image Acquisition and Fundamentals of Digital Image Processing, the face of a user has been detected in real time. By using Face Recognition and Serial data communication, the state of Arduino board pin has been controlled. MATLAB programming develops a computer vision system in the real time for face detection and tracking using camera as image acquisition hardware. Arduino programming provides an interfacing of a hardware prototype with control signals generated by real time face detection and tracking.

**Keywords:** MATLAB, Arduino board, ATmega328P, Digital Image processing, serial data

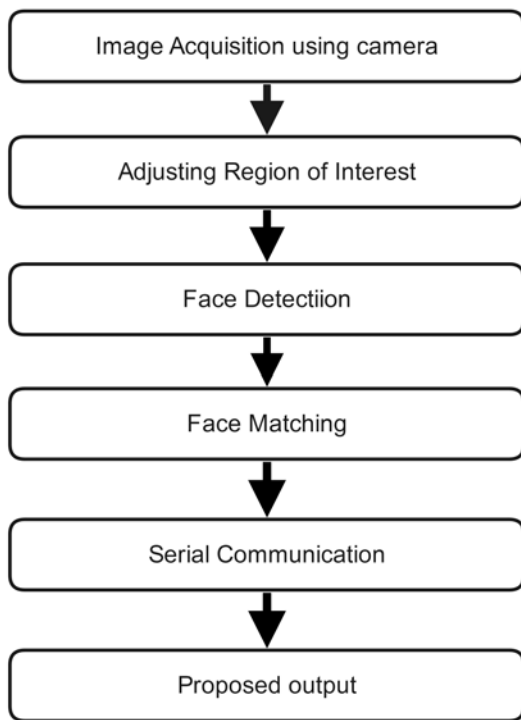
## Introduction

The revolutionized computers open up the chances of using images and video frames as input signals of the signal processing. Such signal processing is named as image processing. Image processing transforms various sets of characteristics of image parameters into output as control signals. The constant revolution in the field of digital image processing opens up a multitude of application in various areas, in which innovative technologies could have been developed. The best platform on which many image processing algorithms have been developed so far is MATLAB. Major advantage of using MATLAB as an image processing algorithm development environment is it's built in image processing functions and its compatibility with hardware such as cameras, Arduino Raspberry Pie and many more.

The aim of this paper is to propose a prototype model which both detects and matches a face with distinct features and generates and sends a control signal to the hardware according to the face recognized. This proposed prototype system contains both software and hardware tools. Software includes MATLAB and Arduino IDE software whereas

hardware includes Camera, Arduino board with ATmega328P microcontroller.

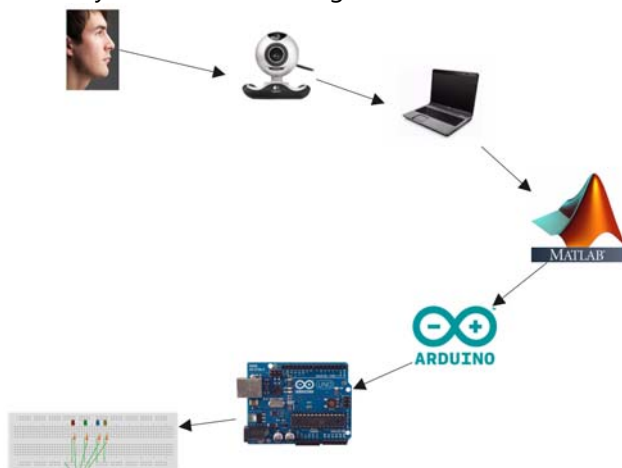
The goal of this work is to visually detect and match a face and send the data to the Arduino board to glow LEDs connected with the microcontroller's digital output port. The most challenging issue rose while developing the face recognition algorithm is that detection of face against complex background. Here, we are using PCA method with Eigen faces to recognize the face. Working of the proposed prototype system can be understood by the following flowchart shown in fig. 1



**Figure1:** Flow chart of proposed system

## PROPOSED SYSTEM

The Face Recognition is done by detecting and matching the face of a user. Here, a hardware set up is required which can be used as an image acquisition hardware such as camera. This camera is connected to a computer. Computer captures the video from the camera. An algorithm for face recognition is written in MATLAB environment. This program recognizes the face of a user in real time. Depending on the face recognized, the control signal is generated and sent through serial communication to Arduino board. State of LEDs connected to digital output pin of ATmega328P microcontroller shows that who is using the computer. Experimental setup of this system is shown in Fig. 2



**Figure 2:** Experimental Setup

## Software Implementation

The entire algorithm for face recognition is based on image processing. The proposed system uses MATLAB as a platform on which image processing algorithm has been developed and tested. As an image acquisition device, camera is used. A camera can be an inbuilt camera of laptop or it can be a USB camera as well. To get the details of the hardware device interfaced with the computer, **imqhwinfo** command of MATLAB is used. Entire MATLAB program for this algorithm can be divided in parts as follows.

### A. Image Acquisition:

The first task in image processing algorithm is to get the live video feed from the camera connected. This live video feed has further been converted into sequence of frames and these frames are used in order to apply further image processing algorithm. For that **getsnapshot** command of MATLAB is used, which converts video feed into image array. Conditionally, ROI (Region of Interest) can also be defined for capturing specified area of the frame. Image acquisition toolbox of MATLAB can also be used for image acquisition purpose.

### B. Face Detection:

Stepping the video frame will make sure we keep getting image input from the webcam and by checking the size of the returning box we will be able to get the number of face detected in the video frame.

- In order to track the face a rectangle box is impended on to the face while tracking.

$$x = bbox(1, 1);$$

$$y = bbox(1, 2);$$

$$w = bbox(1, 3);$$

$$h = bbox(1, 4);$$

$$bboxPolygon = [x, y, x+w, y, x+w, y+h, x, y+h];$$

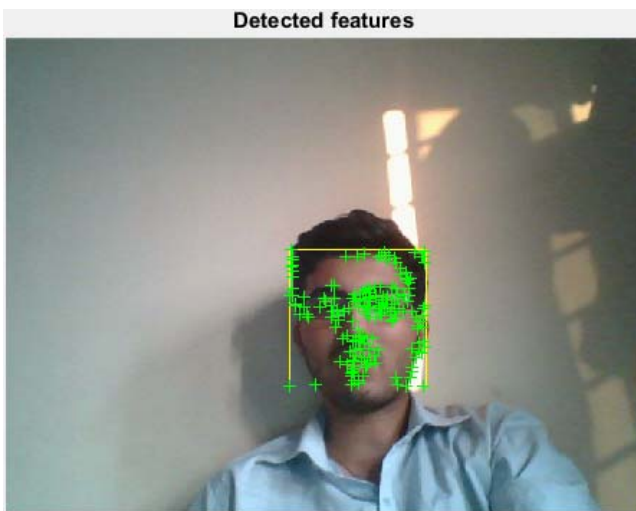
Where  $bboxPolygon(1:2:end)$  represents the x values of the polygon and  $bboxPolygon(2:2:end)$  represents the y Values of the polygon.



**Figure 3:** Face detection using rectangle box

- To detect the points inside of the face region, therefore the following code is going to give us the corner points in face region.

```
Points=detectMinEigenFeatures(rgb2gray(videoFrame)
, 'ROI', bbox);
```



**Figure 4:** Detecting points on the face.

### C. Face Matching:

Principal Component Analysis (PCA) is used to match the image in the database. First a set of data images are stored in the program. After we have to use PCA on the database, the original data will be in the forms of the eigenvectors.

After inputting an image, we will measure the difference between the eigenvectors in input images with the original images in the dataset, and then we need to determine which picture has the least difference to identify the input image. The general eigenvector calculation requires that we subtract the mean of our database.

- $m = \text{uint8}(\text{mean}(v, 2));$  %m is the mean of all images, v is the database.  
 $vzm = v - \text{uint8}(\text{single}(m) * \text{single}(O));$

% vzm is v with the mean removed.

After subtracting the mean from our database set, we will need to find the correlation matrix, and find the eigenvectors to it.

- $L = \text{single}(vzm)' * \text{single}(vzm);$   
 $[V, D] = \text{eig}(L);$

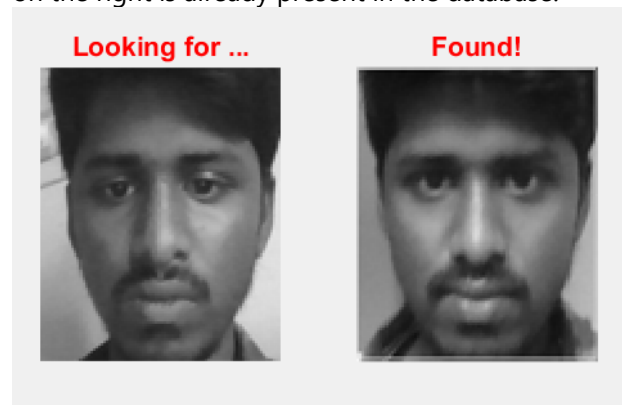
$V = \text{single}(vzm) * V;$

$V = V(:, \text{end}:-1:\text{end}-(N-1));$

%N = 10 which means we are choosing the 10 largest eigenvectors.

As you can see here, V can give us the eigenvectors for L, whereas L is the correlation matrix, ordered by its Eigen values in D from low to high. Single is just converting data to single precision.

The image of the person is captured and the points on the face are detected and it is stored in the database. When a new image of the person is snapped its output may come out to be positive only if the points detected are almost same to the previous detection. Figure 5 shows the two faces one on the left is the currently captured image and image on the right is already present in the database.



**Figure 5:** Matched faces

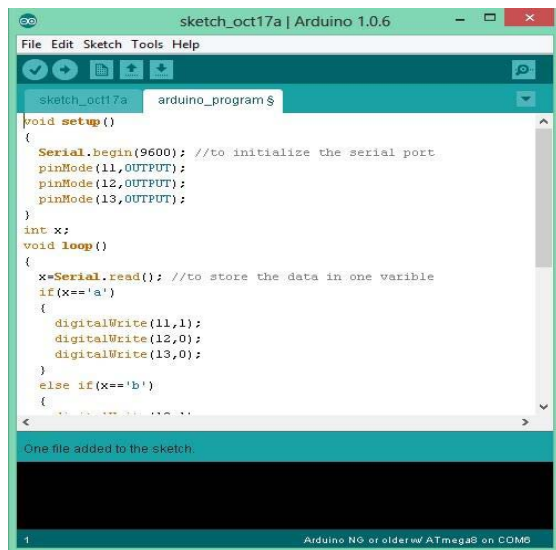
### D. Interfacing MATLAB with Arduino

To interface MATLAB with Arduino, certain Arduino libraries are downloaded in MATLAB IDE, this creates a Arduino to run in MATLAB environment. Communication of hand gesture algorithm with Arduino board is done through serial data transfer. To access the serial port of a computer using MATLAB, couple of lines of coding should be done. MATLAB function for serial port access is shown below.

```
ser = serial('COM18');
```

Here, COM18 is a serial communication port on which Arduino board is connected. Moreover, a

program has to be uploaded to ATmega328P using Arduino IDE. This program makes serial communication port as an input port for the Arduino board.



**Figure 6:** Arduino IDE

## HARDWARE IMPLEMENTATION

Camera is the first and the most important hardware in this system used for image acquisition purpose. Secondly, the implementation of face recognition algorithm on hardware is a challenging task. Proper selection of a hardware is important for the effective working of the system. Here, Arduino board is used which is powered by ATmega328P microcontroller. Arduino Uno can be programmed using Arduino IDE with precise serial port adjustment. Information regarding hardware and their roles in the system is explained below.

### A. Image acquisition device

Various types of cameras are available today. The one with good resolution should be used for the effectiveness of the system. Generally, a laptop in-built web cam or USB camera is used as an imaging device. Here, the device takes continuous snapshots with predefined triggering rate. These continuous snapshots look like a video as that has been taken in very small fraction of time. Image acquisition toolbox is the best tool for the adjustment of parameters of an imaging device used in the system.

### B. Arduino Uno

Arduino is an open-source electronics prototyping platform which can be used to develop interactive objects, taking input from a variety of switches or sensors, and controlling a variety of lights, motors and other physical outputs. Arduino can be programmed

according to the user's requirement and used as a standalone system. The main advantages of using Arduino board are the board is inexpensive, used with cross-platform, programming environment is simple and clear and the software is open source and flexible to modify. The Figure below shows an Arduino Uno which is used in this system.



**Figure 7:** Arduino Uno

## RESULTS

SL NO.	USER FACE	OUTPUT	LED ON
1			RED
2			GREEN
3			BLUE
4			YELLOW

**Table 1:** Results Obtained

After developing the whole system, testing of face recognition algorithm and its communication with Arduino board has been tested. Arduino board is programmed such that when a board receives a character from the MATLAB, digital output pin or a board sets HIGH or LOW accordingly. LEDs are connected on this digital output ports and by receiving the signals from the MATLAB, they changes their states from ON to OFF and OFF to ON. Depending on the face recognized, digital output pins on Arduino board sets up HIGH or LOW accordingly. Table shown above states various outputs.

## CONCLUSION

Main aim of this prototype system is to detect a face, track it, match it with stored Eigen faces and accordingly set digital pin of Arduino board HIGH or LOW. Using MATLAB, face recognition algorithm has been developed with the PCM technique. The Eigen faces are stored first and then we take snapshot of user's face in real time. Then we match the user's face with stored faces and we interfaced this Face recognition with Arduino using Serial communication.

## References

[1] Panth Shah and Tithi Vyas (2014), "Interfacing of MATLAB with Arduino for Object Detection Algorithm Implementation using Serial Communication", International Journal of Engineering Research & Technology (IJERT).

[2] Chunming Li, Yanhua Diao and Hongtao Ma (2009), "A Statistical PCA Method for Face Recognition", IEEE.

[3] V. Subburaman and S. Marcel. Fast Bounding Box estimation based face detection in "Workshop on Face Detection of the European Conference on Computer Vision (ECCV)", 2010.

[4] Raquib Buksh, Soumyajit Routh, Parthib Mitra, Subhajit Banik, Abhishek Mallik, Sauvik Das Gupta, "Implementation of MATLAB based object detection technique on Arduino Board and iROBOT CREATE", IJSRP, Vol. 4, Issue 1, Jan 2014, ISSN: 2250-3153.

[5] Nikhil Sawake, "Intelligent Robotic Arm", Submitted to Innovation Cell, IIT- Bombay, July 2013.

[6] P. Jenifer Martina, P. Nagarajan and P. Karthikeyan (2013), "Hand Gesture Recognition Based Real-time Command System", International Journal of Computer Science and Mobile Computing (IJCSMC).

[7] Viola Paul, J. Jones. Michael, "Rapid Object Detection Using a Boosted Cascade on Simple Features", IEEE CVPR, Vol.1, NO. 2, p511~518, Dec. 2001.

[8] Gary Bradski, Adrian Kaehler, Vadim Pisarevsky, "Learning-Based Computer Vision with Intel's Open Source Computer Vision Library", Intel Technology Journal, Vol 9, Issue 2, p119~131, 19 May 2005.

[9] <https://www.mathsworld.com>.

[10] <https://www.arduino.cc>.

## Author's details

1 Student, Computer Science Department, Dayananda Sagar Academy of Technology and Management, Karnataka, India, kirana.r.mankale@gmail.com

2 Student, Computer Science Department, Dayananda Sagar Academy of Technology and Management, Karnataka, India, mlohith4u@gmail.com

3 Student, Computer Science Department, Dayananda Sagar Academy of Technology and Management, Karnataka, India, eyogiyogesh@gmail.com

4 Student, Computer Science Department, Dayananda Sagar Academy of Technology and Management, Karnataka, India, asaikumargowd@gmail.com

5 Professor, Computer Science Department, Dayananda Sagar Academy of Technology and Management, Karnataka, India, anitha.jayapalan@gmail.com