Project_Report_draft_v1.6.docx

WORD COUNT	6633	TIME SUBMITTED	14-MAR-2023 09:46PM
		PAPER ID	97609941

Analysis & Development of an Android-based Face Recognition System

A Project Report Submitted to the
Department of Computer Science and Engineering, Jahangirnagar University
in Partial Fulfilment of the Requirements for the Degree of
M.Sc. in Computer Science and Engineering

Prepared By

MD. MAHBUBUR RAHMAN

Student Id No.: CSE202201013

Registration No: 2901

Enrollment: Spring-2022

Batch: 28th

Under Supervision of

SARNALI BASAK

Associate Professor

Department of Computer Science and Engineering
Jahangirnagar University

MARCH 2023

Approval of Acceptance

The project report entitled "**Analysis & Development of an Android-based Face Recognition System**" is written and submitted by **Md. Mahbubur Rahman**, Student ID No. **CSE202201013** to the PMSCS program, Department of Computer Science and Engineering, Jahangirnagar University in partial fulfillment of the requirements for the degree of Master's in Computer Science. This project is done under the supervision of **Ms. Sarnali Basak, Associate Professor,** Department of Computer Science and Engineering, Jahangirnagar University.

We have examined this report and recommend its acceptance.

.....

SARNALI BASAK
Associate Professor
Supervisor & 1st Examiner
Department of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

.....

DR. MD. EZHARUL ISLAM
Professor
2nd Examiner
Department of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

Accepted by:

.....

DR. MD. HUMAYUN KABIR
Professor
Coordinator, PMSCS Program
Department of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

Abstract

Smart phones have been come up with a fast evaluation and amazing features in the last decade. Now a days, a smart phone is equipped with the components which are going to replace so many daily use electronic devices like Camera, Wristwatch, Calculator, Compass even a whole Personal Computer. After formation of 'Open Handset Alliance', Android operating system became the game changer in smart phone industry. A huge number of android apps are being developed and made available in play-stores to meet different daily requirements as well as for fancy purpose. Most android apps have free or demo version and paid version and obviously the paid versions are come in useful for real life. A few years back, CC camera systems are supposed to be an essential security measure for a VIP area or a restricted place. But now a days, it has become as common as other appliances for personal use at home. Google play stores and other app stores have vast collection of apps to use on this purpose but each and every app has a major limitation of object detection and face recognition that is the part on which this project is mainly focused. One can easily ensure surveillance of a house or an office or any restricted place where 24/7 surveillance is needed through identifying any intruder or tracking down the entries of undesired persons with a PC or laptop browser and obviously with an android smart phone without any cost.

Declaration

The project work with title "**Analysis & Development of an Android-based Face Recognition System**" has been done in the Department of Computer Science & Engineering, Jahangirnagar University is genuine and compliant to the regulations of this University.

I do understand the plagiarism policy of this University and declare that this project is a unique work and no part of it has been plagiarized from other sources or been submitted previously elsewhere for awarding any degree, diploma, or publication.

(Md. Mahbubur Rahman)
Roll No. – CSE202201013

Counter Signed by

SARNALI BASAK
Associate Professor & Supervisor

Acknowledgement

I am extremely grateful and remain indebted to Al-mighty Creator who has guided in all. Ventures to successfully complete my project. I am thankful to the grace and the help received from him. Special thanks go to our honorable supervisor **MS. SARNALI BASAK**, Associate Professor, Department of Computer Science & Engineering, Jahangirnagar University. I would address with immense gratitude to the cordial support of my supervisor whose instructions, advice and sterling accent perfectly guided me towards accomplishment of this project work. Besides, I would like to thank all my course teachers who were so kind upon me in evaluation for last three semesters of PMSCS program. Undoubtedly, we are under tremendous supervision of our Honorable PMSCS program Coordinator **DR. MD. HUMAYUN KABIR**, Professor, Department of Computer Science & Engineering, Jahangirnagar University. I am bound to acknowledge with heartiest thanks to all other faculty members of Department of Computer Science & Engineering, Jahangirnagar University for their valuable time spend to analyze and evaluate my project.

In addition, I would recall the memories of intimate cooperation and support from friends and family and the people of my surroundings who contributed directly or indirectly for successful completion of this project. I am truly in a debt to them for encouraging me to carry on the project works even in my hardest times. I am also grateful to the people who criticized my works and helped to identify and resolve errors in my working procedure and methodology in several phases during development of this project and in report writing as well.

Heartiest thanks to my batch mates who have contributed directly through their valuable opinions for the development work and its associated activities.

Contents

Chapter 1

Introduction

Object detection and face recognition systems are considered as the ultimate application of 'Digital Image Processing' system. Machine learning, Artificial Intelligence added a new extent in this task as well. Technology gave the opportunity to mankind to do something better beyond their own thinking through these amazing technologies. Object detection and face recognition can be done in basic two manners – from still images that means a single frame of image and other is from a video stream which is actually consists of several frames in a second. The last one is comparatively complex since the particle image is identical in different frames and to recognize the object there should be plenty of samples to compare with the original. As the number of samples increases, the system can detect and recognize the object more precisely.

This project mainly focused on second category of object recognition especially human faces. Face pattern is one of the most prominent identifications of a person and as a social being we are introduced first with our faces. We memorize, recognize and interact with a person with his/her face. Even for an effective surveillance system, nothing extra to say about the importance of detecting intruders. In a conventional CC Camera system, a person should watch the streaming of video captured by the cameras placed in vital observation point of a premises and the watcher should be alert all the time on screen to detect any intruders or anomalies. Market available CC camera systems are capable to detect motions and some advanced system can perform some operations like warning alarms, capturing still pictures and sending SMS/Email Notifications etc. based on motion detection only. This project offers a better alternative solution for this conventional surveillance system. It can stream the camera captured video, detect human faces and moreover recognize the faces according to prerecorded images. Thus, it can easily identify a person who stands before the camera to enter a premises and alert security individuals about the intruders.

From the beginning of this century smart phones emerged as the most dominating electronic devices in our daily life. Its small size, portability, ability to meet up the requirement of other common devices like Camera, Watch, Calculator, Compass etc. facility made this device a companion for round the clock of our daily life. So, it will be very convenient to operate a surveillance system using a mobile app operated through Android driven smart phones.

1.1 Objective

To build a CC Camera surveillance system which is accessible and operated from the interface of Android App and has the capability not only to detect human faces but also to recognize a person. Developing a simple mobile interface with secured logging in with credentials, customizable Wifi LAN IP, human face registration and recognition. It is believed to have most of user-friendly features, satisfactory performance and obviously well secured. Most importantly, all these facilities will be available at a very low cost.

1.2 Goals

Technology is now growing faster than our thinking; we are going to enter a new AI based era. There are already lot more integration of modern technologies like machine learning and real time video analysis techniques in such processing. Not only object detection and face recognition, machines can now understand different human gestures and specific body movement. However, this project will facilitate a complete solution of surveillance system which can be bult with cheap and available electronic circuitries and software components. And it will be usable through most common appliance of our daily life like Mobile phones, PC and laptops, Android TVs. It may be considered as an effort to initial step for adoption of modern sophisticated technologies in our everyday life.

Chapter 2

Literature Review

Digital Image processing facilitates detection of different objects as well as human face recognition. This facility lets to build machine-based person identification system by recognizing an individual's face. Applications of face recognition are not only limited in security purposes but also in various of fields like Payment Card Industry, e-Business, Mobile Applications, analysis of weather, and climate, Multimedia Applications etc. Face recognition methods are divided into two basic types - based on Appearance and based on Model. Face recognition based on appearance means capturing the full face of a person and analyzing pixel information to recognize identical faces. Face recognition based on model specifies the procedure of constricting a model of the person's face with different expressions. Face images are extracted to their corresponding vectors and similarity between two images is determined through derivation of the distance between those image vectors. The smallest distance indicates the match case of captured images to the pre-stored images. The term Recognition addresses the identity of the match case image.

2.1 Generic System of Face Recognition

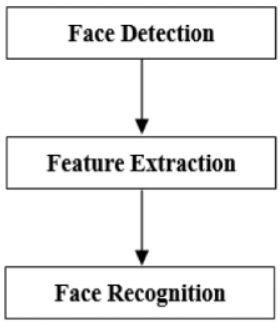


Figure: Generic System of Face Recognition as shown in [2, Fig. 1]

2.2 Face Recognition Method Based on Appearances

In this method, faces are identified using full face image instead of local specification of face that means position and shape of the significant parts of the face. Global facial structure is constructed directly from the combination of pixels of face images which identically capture the variation of different person faces and used to identify those persons. The face is captured in such way that entire face must exist in the image frame and this frame is later used as an input for the system to recognize that person's face.

2.3 Face Recognition Method Based on Models

Numerous image samples are captured first to construct a model of an individual's face. The sample images are taken by capturing the face from different angles and facial gestures of the person. Later this model is used to recognize the persons' face by measuring the distance and position of major facial parts i.e., eye, nose, lips, chin etc. Main facility of this method is it performs better with aby size, orientation, and poor lighting condition. Compactness is another plus point of this method which can make faster recognition.

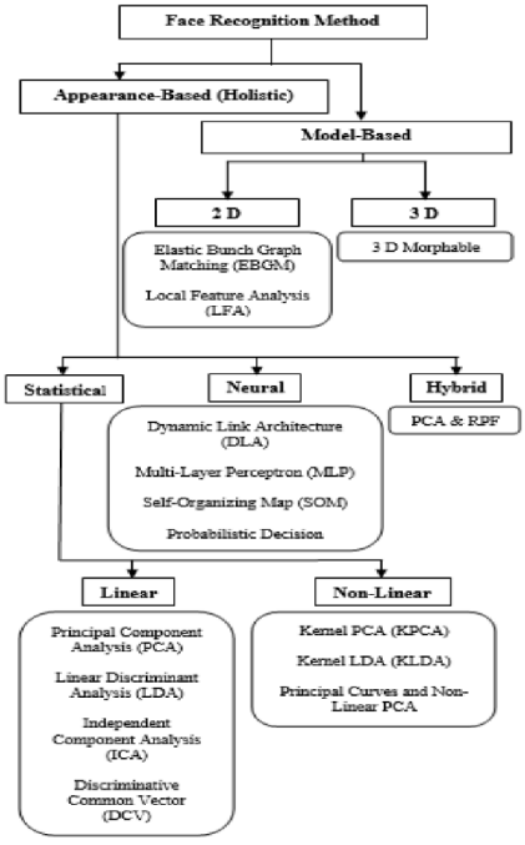


Figure: Face Recognition Methods as shown in [2, Fig. 2]

2.4 Distance Measurement

Almost each face recognition method uses the technique of distance measurement between two images. This measurement is the main key to recognize a face. Another term is used to express this fact and that is 'Image similarity' which refers to the difference between image vectors. Below a short overview is given on different distance measurements.

Euclidean Distance

A widely used distance measuring technique and faster than other classifiers because it is too simple. Euclidean Distance refers to the distance calculated from straight line distance between two points which is derived from square difference between the coordinators of two different objects.

Square Euclidean Distance

The major difference between Square Euclidean Distance and Euclidean Distance is, it can be calculated without the square roots.

Some other renowned distance measuring methods - City Block Distance, Chebyshev Distance, Hausdorff Distance, Mahalanobis Distance etc.

2.5 Research Purpose Face Recognition Database

There are several face databases available for the researchers who are conducting various research works on face recognition. Researchers select the database depending on their purpose of research since those databases are different in size and specifications. The pictures preserved in those databases are captured by numerous small researcher groups on their interests and study of face recognition.

Some of the popular face recognition databases are listed on the following page.

Name	Individuals	Image Resolution
AT&T (ORL)	40	92 x 112 Gray Scale
FERET	1199	256 x 384 Gray Scale/ RGB
AR Face Database	126 70 Male 56 Female	576 x 768 RGB
PIE Database, CMU	68	640 x 486 RGB
BioID Face Database	23	382 x 288 Gray Scale
The Yale Face Database	15 14 Male 1 Female	320 x 243 Gray Scale
The Yale Face Database B	10	640 x 480 Gray Scale
UMIST Face Database	20	92 x 112 Gray Scale
The MUCT Landmarked	276	480 x 640 RGB
Face 94	153	180 x 200 RGB
Indian Database	40	640 x 480 RGB
Grimace Database	20	180 x 200 RGB

Face Recognition Database Features as seen in [2, Tab. 2]

Chapter 3

Methodology

Face Recognition is one of the most popular biometric authentications that has a wide range of applications. Apart from security surveillance systems, it has real life use in digital attendance system and other multimedia purposes. The methodology of a face recognition system entirely depends on the architecture of its basic functionality, hardware devices by which the system will be built and finally the user interface environment and its development platform. The working principle of a face recognition system can be divided into three parts – System Train up, Detection & Recognition. First the system is trained up with some sample images of a person and store the sample images in memory. In detection part the system identifies the face of an individual's image based on the position of facial parts. Finally in recognition part, the system calculates the distances between stored image vectors and the live image vector, or a frame of video stream and determines the matched image sample of the person.

There are three segments in this project,

1) Hardware Segment
2) Firmware Segment
3) Software or Application Segment.

3.1 Hardware Segment

This segment is all about an IP Camera module known as **ESP 32 CAM** and its interface Arduino Board. ESP 32 CAM board is small in size and contains a 2 mega pixel Camera MODEL: OV2640. The board can communicate with the network through Wi-Fi and Bluetooth and can be configured with varieties of settings for video streaming. It is operated either 3.3V or 5V power input. It is equipped with a bright LED to take photos in low light condition. A micro-SD memory card slot is also available on it which can support 4 GB memory cards to store images.

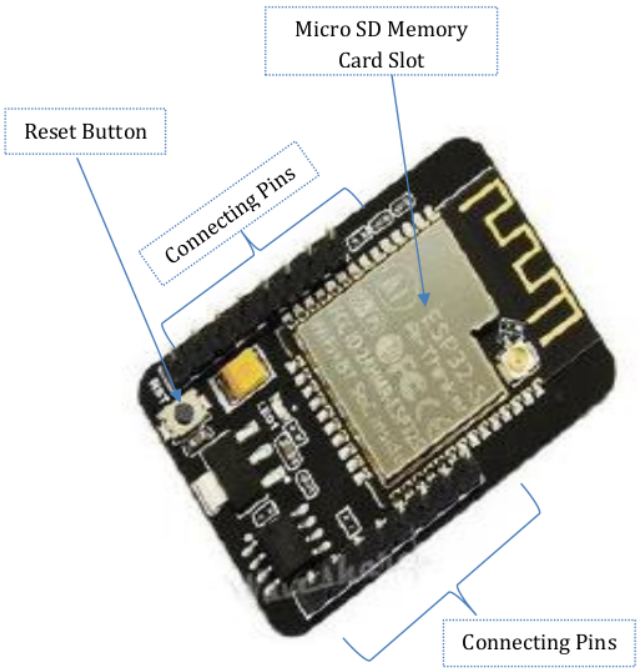


Fig 3.1: Front Side of ESP 32 CAM Board

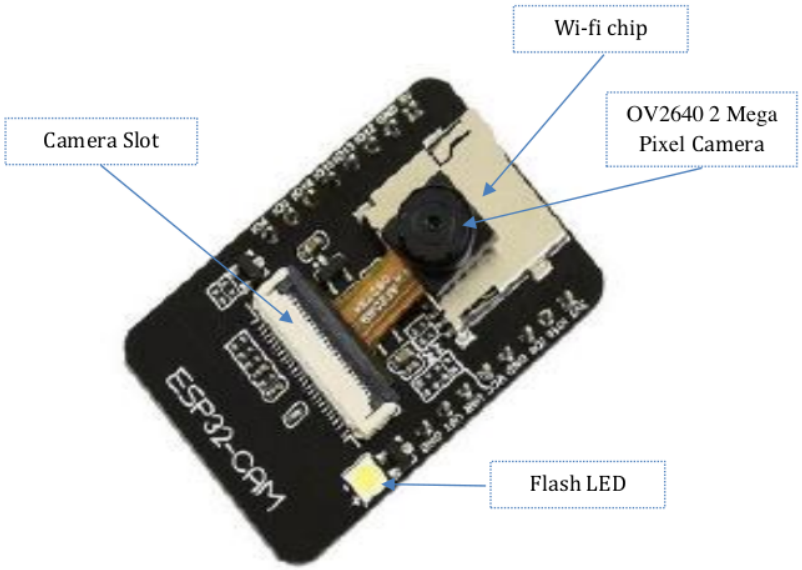


Fig 3.2: Back Side of ESP 32 CAM Board

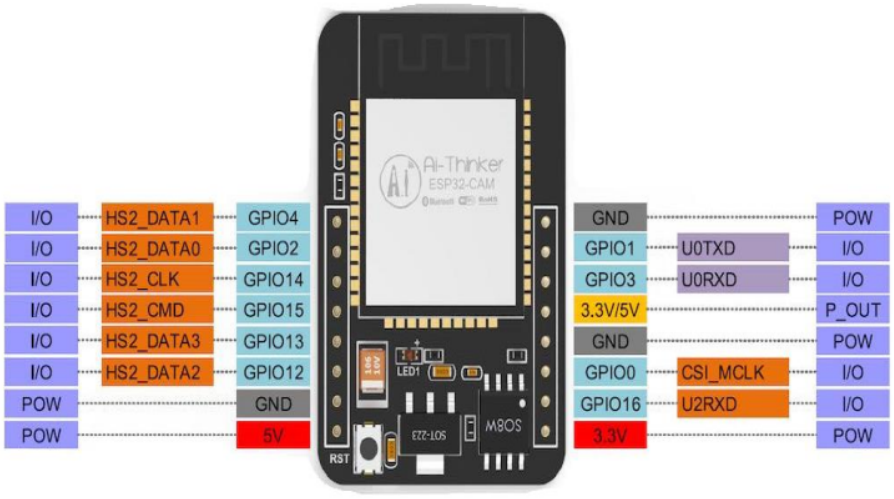


Fig 3.3: Pin Configuration of ESP 32 CAM Board

Communication Interfacing Arduino Board: To make the ESP 32 CAM board workable, an operating code must be burned in it. This task is done by Communication Interfacing Arduino Board. This board works as a controller and a converter.

Fig 3.4: Interfacing Arduino Board

The board is simpler and cheaper and very handy to use. It is easily accessible as well.

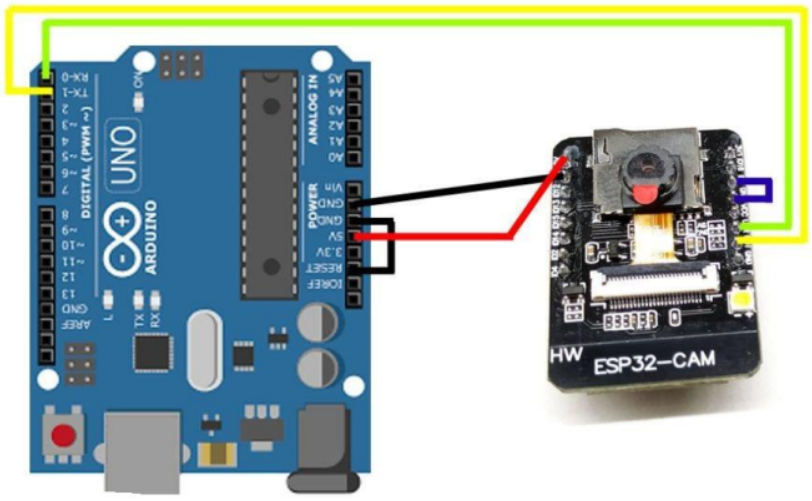


Fig 3.5: Wire Connection between ESP 32 CAM & Arduino Board

To burn codes in the ESP 32 CAM board, it should have wire connection to the Arduino board according to the figure above. After the wire connections are done, necessary code can be uploaded to the board from a PC or laptop. A software interface Arduino IDE is also required to this job. Firmware codes can be created, edited, updated, and uploaded from that IDE Software module. For connectivity to a Wi-Fi LAN, necessary credentials like SSID and password of the network may be hard coded in the firmware code.

Access of Camera

After compiling and uploading the firmware code, the ESP 32 CAM board needs to be reset and then it will be connected to desired Wi-Fi LAN. In serial monitor of Arduino IDE code uploader software, an IP address will be shown which is assigned to the camera by Wi-Fi router. This IP address will be used by user applications as the URL to access its functionality.

3.2 Firmware Segment

An operating code is created and uploaded in ESP 32 CAM board which includes necessary header files, functions and pin definitions. For this project, the corresponding significance of firmware works is much higher than other methodologies because, main procedure of face detection and recognition process is done here. That means, several samples of a person's face are first taken and saved after conversion to RGB vector. In detection part the saved vectors are compared with the captured image of a new face. The identity of the closest matched image vector is recognized as the ident of the new face.

To do the firmware job, let's be introduced with the Arduino IDE interface and know the procedure work with ESP 32 CAM board.

Fig 3.6: Arduino IDE

The Arduino IDE is an open-source project published by ARDUINO.cc. It is a great platform to work with verities of programmable chips and boards. It provides a user-friendly interface to write programs in a high-level language programming paradigm almost like 'C'.

Initial Setup of Arduino IDE: When the wire connection between ESP 32 Cam board and the Arduino board is done, the Arduino IDE must be configured to write codes and upload to the ESP 32 cam board microcontroller chip. Generally, a fresh installation of Arduino UDE program does not include the ESP 32 board manager, it has to be downloaded from respective URL. The URL needs to be specified in the 'Preference' settings of the software as shown below.

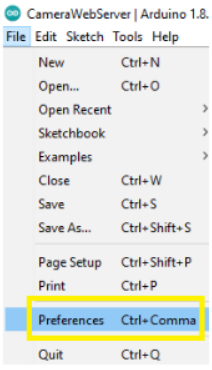


Fig 3.7: Preference Menu Option in Arduin IDE

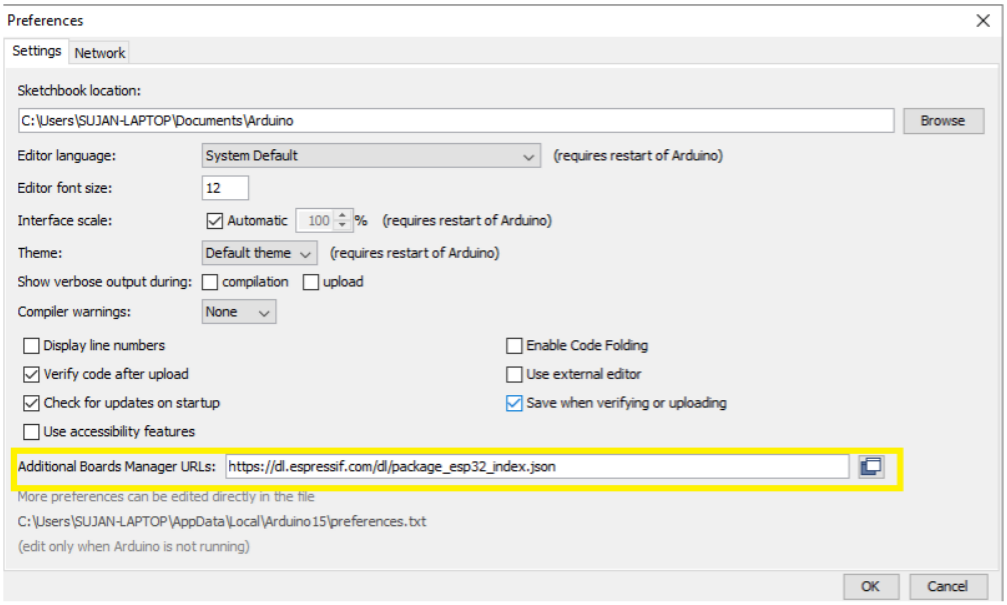


Fig 3.8: Preference Settings for ESP 32 Board Manager Library

Now, the Arduino IDE needs to be restarted and the board should be connected to USB port. Then the ESP Library needs to be installed from Tools Menu → Board → Board Manager → typing search key "ESP 32" → selecting the search result and click install.

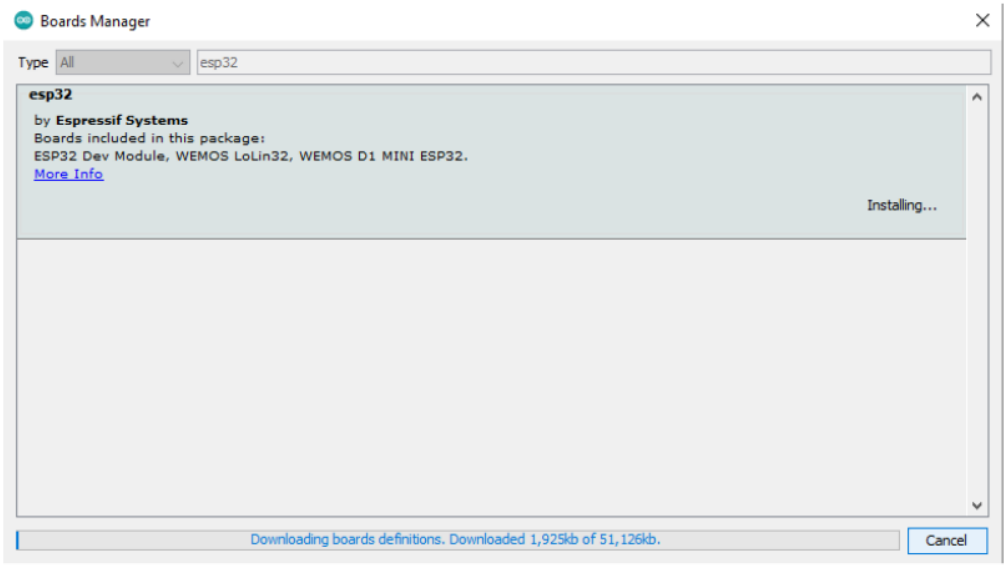


Fig 3.9: Installing ESP 32 Library files in Board Manager

The following specifications and settings should be set before writing codes in Arduino IDE. These parameters will be available in '**Tools'** menu.

- Board Name : ESP 32 AI THINKER
- Upload Speed : 115200
- Partition Scheme : Minimal SPIFFS
- Port : Serial Port COMX (X is the USB port no to which the Arduino Board is connected)
- Flash Mode : QIO
- Flash Frequency : 80 MHz
- Core Debug Level : Debug

The 'serial monitor' in tools menu should be kept enabled to view the code upload status and diagnosis the errors during code burning to the board's chip.

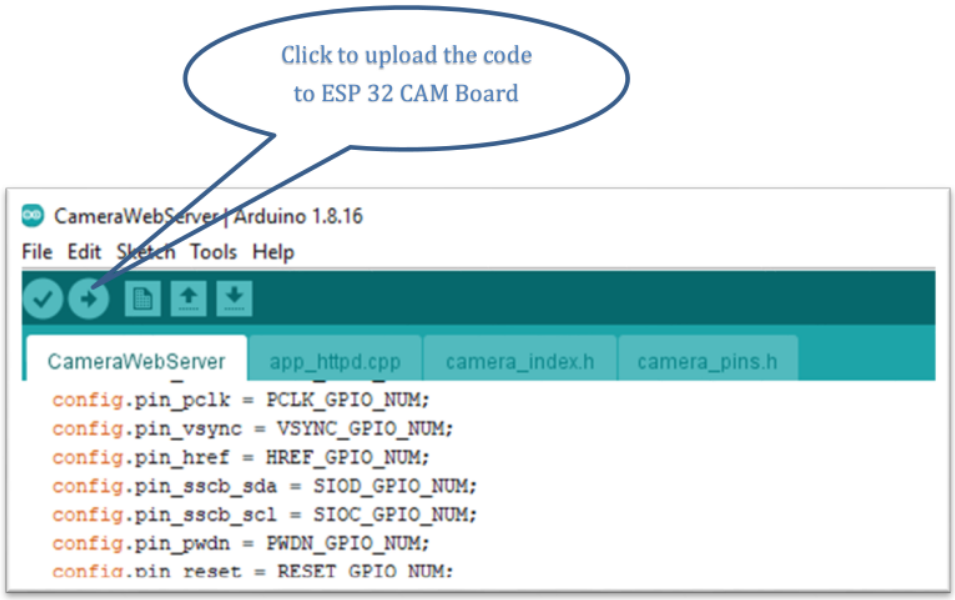


Fig 3.10: Uploading code from Arduino IDE to ESP 32 CAM board.

To create the firmware for this project, at first necessary code should be written to make the ESP 32 CAM board **a Camera Web Server**. This job can be done also by modifying a given examples with Arduino IDE. A file with ***.ino** extension is created when the code is saved. Besides three more files are created with *.cpp & *.h extensions. These files contain the definition of all functions and header files. The **camera_index.h** file contains the RGB mapping of the camera pixel and the **camera_pins.h** defines pin configuration for different settings.

Finding the Camera IP Address: When the code upload process is completed, the GPIO pin should be disconnected from Arduino Burner device and the ESP 32 CAM board's 'Reset' button needs to be pressed. Then, the board will be restated and get connected to Wi-Fi network and in the serial monitor IP address will be shown.

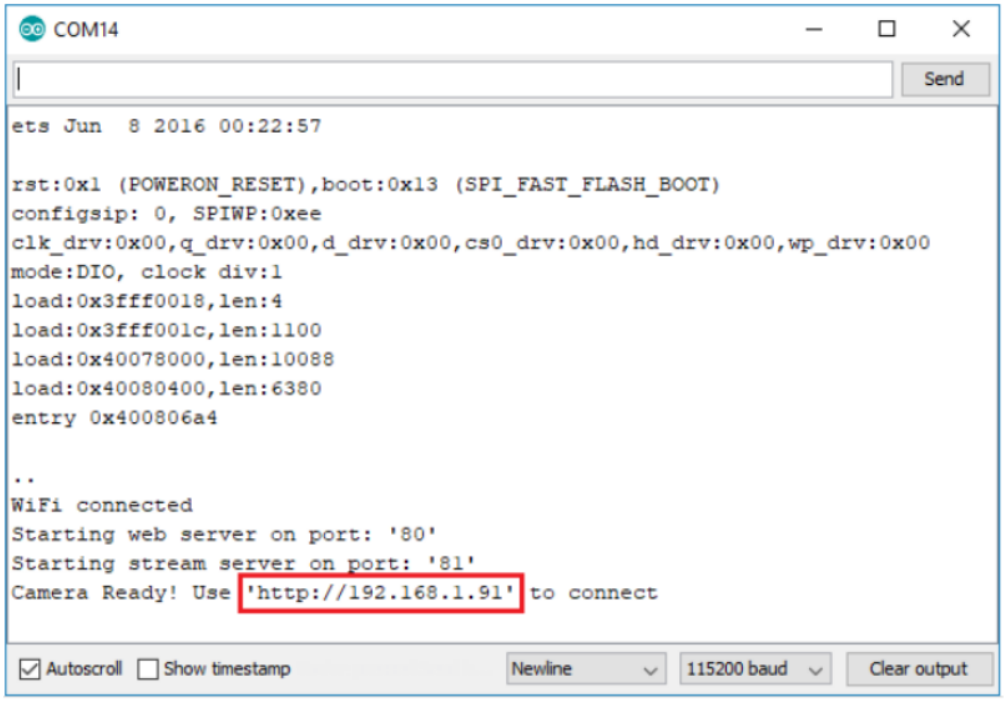


Fig 3.11: ESP 32 CAM board Ready State and Obtaining IP address.

Brief Focus on Firmware Files: The most significant firmware files used for this project is discussed below with their roles, specifications and functionality in brief.

……..…..……**Firmware File -1**……

File Name : *CameraWebServer.ino*
Role : *To contain firmware code written in high level language paradigm*
Specification :

- Header files : esp_camera.h, WiFi.h, camera_pins.h
- Camera Definition : CAMERA_MODEL_AI_THINKER

- Pin Configuration :

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
```

- Major Functions :

Function Name	**Activity**
defined()	Defining input pins
esp_camera_sensor_get()	Vertical Flip, Brightness & Saturation Control
WiFi.begin(ssid,password)	Connecting to WiFi network
startCameraServer()	Starting Camera Video streaming

……...……**Firmware File -2**……

File Name : *app_httpd.cpp*
Role : *To contain all http handler functions for ESP 32 CAM various operation*
Specification :

- Header files :

```
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "fd_forward.h"
#include "fr_forward.h"
```

- Image Sample Number Settings:

```
#define ENROLL_CONFIRM_TIMES 5
```

- Number of Individual Faces Settings:

```
#define FACE_ID_SAVE_NUMBER 7
```

- http Handler Functions:

http handler Function Name	Activity
led_handler()	.uri → "/led" HTTP_GET method for controlling the on-Board Flash LED
video_handler()	.uri → "/stream" HTTP_GET method for streaming the Camera Video
detect_handler()	.uri → "/detect" HTTP_GET method for enabling/ disabling Face Detection
enroll_handler()	.uri → "/enroll" HTTP_GET method for enabling/ disabling Face Registration
recog_handler()	.uri → "/recog" HTTP_GET method for Enabling/ disabling Face Recognition

- URL formation with the http handlers:
  Example: http://192.168.1.91/stream → shows the camera video Stream

……...……**Firmware File -3**……

File Name : *camera_index.h*
Role : *Contains Byte array in Hex Form to be converted into html file*
Specification :

```
#define index_ov2640_html_gz_len 4316
const uint8_t index_ov2640_html_gz[] = {
 0x1F, 0x8B, 0x08, 0x08, 0x50, 0x5C, 0xAE, 0x5C, 0x00, 0x03, 0x69, 0x6E, 0x64, 0x65, 0x78, 0x5F,
 0x6F, 0x76, 0x32, 0x36, 0x34, 0x30, 0x2E, 0x68, 0x74, 0x6D, 0x6C, 0x00, 0xE5, 0x5D, 0x7B, 0x73,
 0xD3, 0xC6, 0x16, 0xFF, 0x9F, 0x4F, 0x21, 0x04, 0x25, 0xF6, 0x34, 0x76, 0x6C, 0xC7, 0x84, 0xE0,
 0xDA, 0xE2, 0x42, 0x08, 0xD0, 0x19, 0x5E, 0x25, 0x2D, 0x74, 0xA6, 0xD3, 0x81, 0xB5, 0xB4, 0xB2,
 0x55, 0x64, 0xC9, 0x95, 0x56, 0x76, 0x52, 0x26, 0x9F, 0xE3, 0x7E, 0xA0, 0xFB, 0xC5, 0xEE, 0xD9,
                                          .
                                          .
                                          .
                                          .
 0x0B, 0x10, 0x65, 0x5B, 0x49, 0xDA, 0x6C, 0x9E, 0x8F, 0x8A, 0xB0, 0xB8, 0xA8, 0x72, 0xE2, 0x33,
 0x38, 0x8D, 0x36, 0x2C, 0xC5, 0x37, 0xF1, 0x52, 0xAE, 0xC1, 0xA9, 0xF8, 0x9F, 0x0A, 0xFF, 0x0B,
 0x9B, 0xFC, 0x8E, 0x51, 0xC1, 0x70, 0x00, 0x00
};
```

3.3 Application Segment

The application segment includes the development of a user interface either for a PC or a smartphone. The ESP 32 Cam is so well equipped that it can be configured for a user interface that needs only a browser. But advanced utilization of its functionality through a standard user interface requires higher level development platforms like Visual Studio with Python language (for PC applications), Android Studio with JAVA or KOTLIN language (for mobile apps). There is another open-source software development kit called 'Flutter' which can be used for development of mobile app in a very convenient way. In this project, all the user interfaces developed for an android device is developed using Flutter and activities are defined in 'DART' language.

Android Studio is a vast development platform, this project has been worked out with Android Studio bumble bee version. The running version is Android Studio Electric Eel 2022.1.1.

Fig 3.12: Android Studio

Dart SDK is an efficient tool to build high-performance mobile apps. It mainly focused on low-latency input and high frame rates on all platforms. In this project, flutter version 3.7.5 has been used for building the user interface for an android app.

With a fresh installation of Android Studio does not include Flutter or The Dart SDK; both plugins have to be enabled from the settings/configuration option of Android Studio.

3.4 Other requirements

1) One or more than one Smart Phone with standard Android version
2) A Wi-Fi Router for home use
3) Bread Board and
4) Some Connecting wires

3.5 Basic functionality

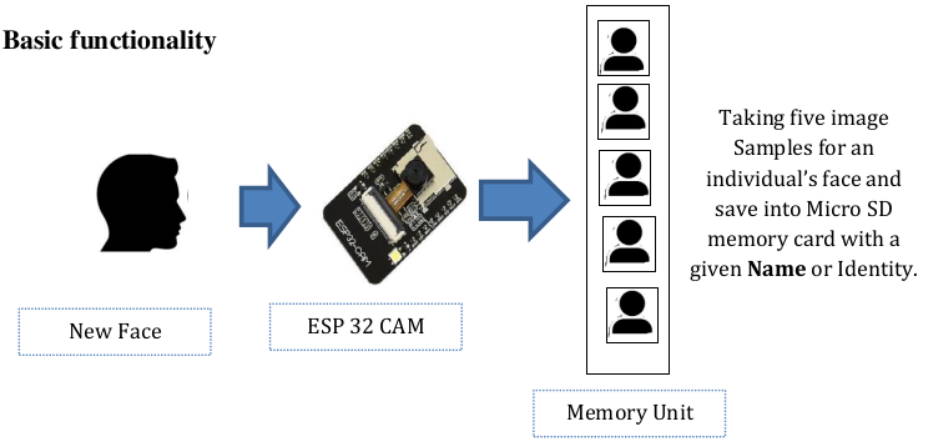


Fig 3.13: Face Registration Process

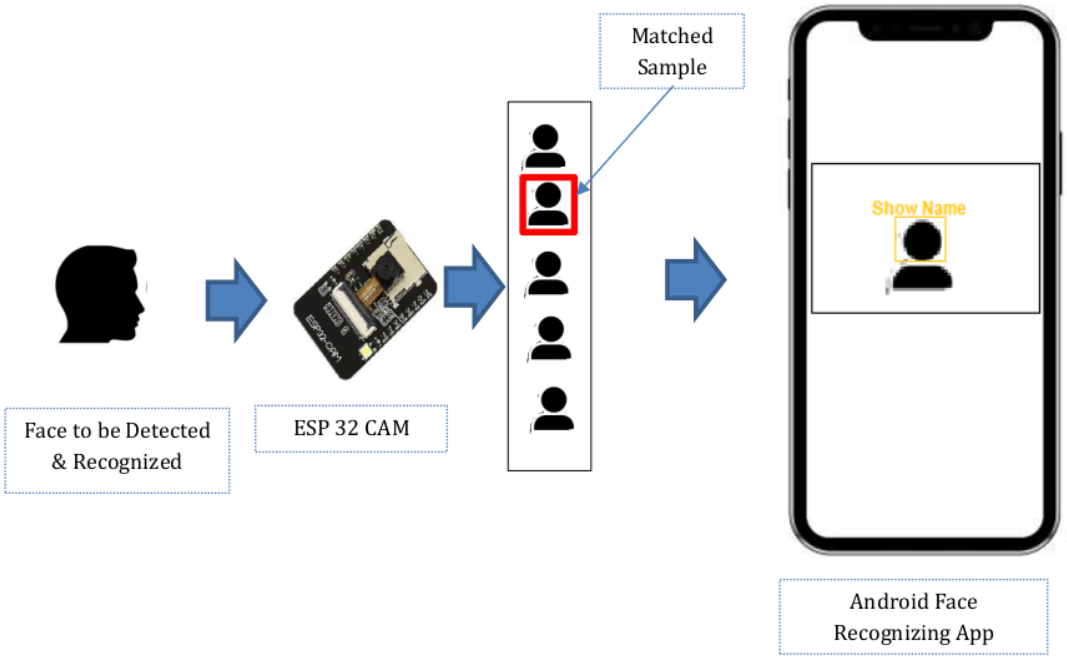


Fig 3.14: Recognizing a Registered Face

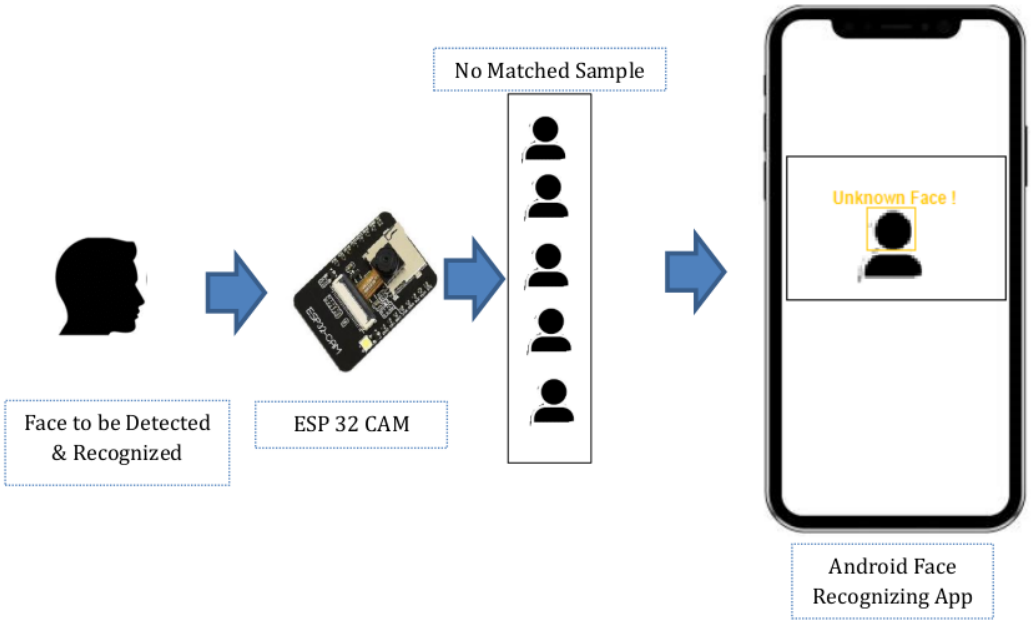


Fig 3.15: Recognizing an unregistered face

N.B. – Since there are 05 sample images have been taken while face registration, a registered face may be detected as unknown face if the face appears in a different angle that does not match with the sample images. That means, the accuracy of face recognition can be increased by taking more sample images from different angles and with different gestures.

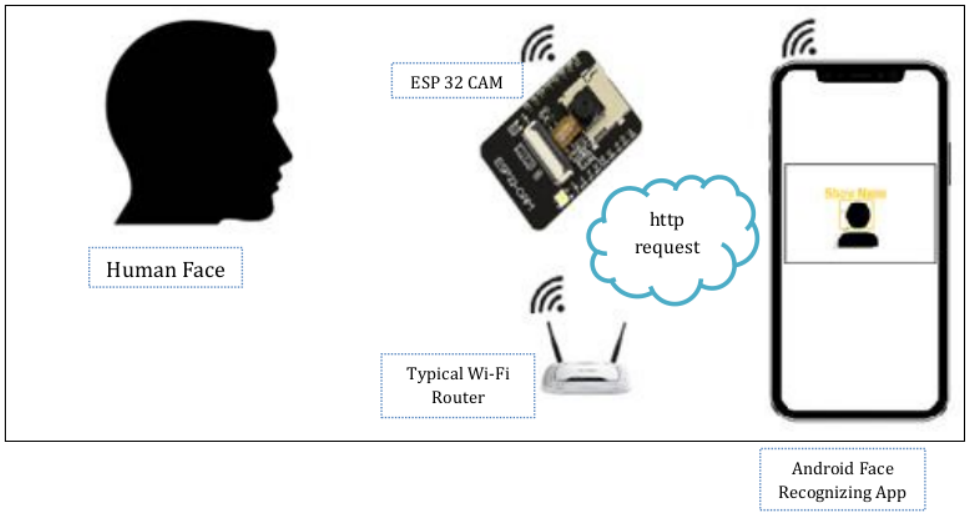


Fig 3.16: Basic architecture/functionality of the project

3.6 Development of the Android App

Basic functional design and steps relating to this project have been discussed earlier; the firmware work of ESP 32 CAM board is considered done. Now, this chapter's articles will go deeper towards building the project. To do that, some sketches may be designed first for the mobile application development and then necessary coding will be done according to the object views of the layouts and their activities. Step by step procedure is discussed in detail below.

Step-1: Creating a new project with the help of 'Flutter plugin' and 'Dart SDK' in Android Studio software. Configure Android Manifest file for desired layouts and activities, main.dart and the video streaming configuration file for the application. Relevant file specifications, role and significant scripts portion are discussed below.

File Name : *AndroidManifest.xml*
Role : *To contain settings, configurations & resource permissions for mobile app*
Significant Script:

- Required Permissions for Resource use:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

.....

File Name : *main.dart*
Role : *Root of the application*
Significant Script:

- Imported packages:

```
import 'package:flutter/material.dart';
import 'Pages/face_recog.dart';
import 'Pages/face_registrer.dart';
import 'Pages/hompage.dart';
import 'Pages/loginpage.dart';
import 'Pages/watch_stream.dart';
```

- Imported dart files:

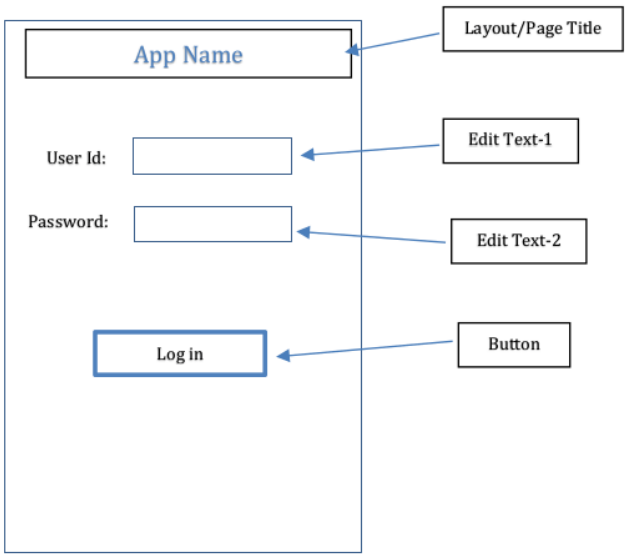
```
import 'package:flutter/material.dart';
import 'Pages/face_recog.dart';
import 'Pages/face_registrer.dart';
import 'Pages/hompage.dart';
import 'Pages/loginpage.dart';
import 'Pages/watch_stream.dart';
```

- Route Definition for different layout pages:

```
routes: {
  'lp': (_) => const LogInPage(title: 'Face Detecting and Recognizing App'),
  'hp': (_) => const MyHomePage(title: 'Homepage'),
  'ws': (_) => WatchStreamPage(title: 'Watch Stream', ip: ''),
  'freg': (_) => FaceRegisterPage(title: 'Face Register', ip: ''),
  'frecog': (_) =>
      FaceRecognitionPage(title: 'Face Recognition Page', ip: ''),
},
```

.....

Step-2: Create a layout for Log in page according to the following sketch.



File Name : *loginpage.dart*
Role : *Definition of Object Views as scaffold widget under LogInPage class and their activities for Login page*

Significant Script:

- Imported package:

```
import 'package:flutter/material.dart';
```

- Default Id-Password setting:

```
class _LogInPageState extends State<LogInPage> {
  int _counter = 0;
  final TextEditingController tEC1 = TextEditingController(),tEC2 = TextEditingController();
  final List<String> userid=['masud','mahbub','system'];
  final List<String> userpsw=['123','123','abc123','abc123'];
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
```

- Edit Text-1:

```
  mainAxisAlignment: MainAxisAlignment.start,
  children: <Widget>[
SizedBox(height:50,width: 300,),
    SizedBox(
      height:50,width: 300,
    child: TextFormField(
      controller:tEC1,
        style: const TextStyle(
          fontSize: 18,
          color: Colors.black,
        ),
        decoration: const InputDecoration(hintText: "User ID",hintStyle:TextStyle(

          fontSize: 18,
          color: Colors.grey,
        ))
    ),
    ),
```

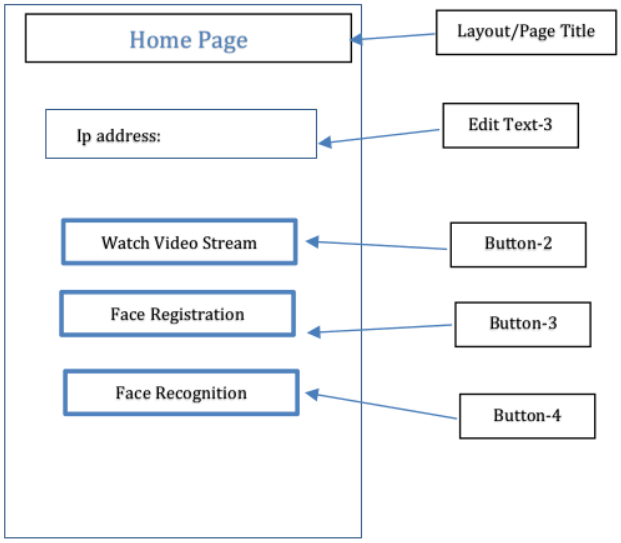
- Edit Text-2:

```
  child: TextFormField(
    controller:tEC2,
    style: const TextStyle(
      fontSize: 18,
      color: Colors.black,
    ),
    decoration: const InputDecoration(hintText: "Password",hintStyle:TextStyle(
      fontSize: 18,
      color: Colors.grey,
    )),
    obscureText: true,
  ),),
```

- Button-1:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5, // foreground
    ),
      onPressed: (){
        if(userid.contains(tEC1.text) && userpsw[userid.indexOf(tEC1.text)]==tEC2.text)
          {
            Navigator.of(context).pushNamed('hp');
          }
        else{
          Fluttertoast.showToast(
              msg: "Invalid User ID/Password",
              toastLength: Toast.LENGTH_SHORT,
              gravity: ToastGravity.BOTTOM,
              timeInSecForIosWeb: 1,
              backgroundColor: const Color(0xFF82A30A),
              textColor: const Color(0xFF0D1141),
              fontSize: 16);
        }
      },
```

Step -3: Create a layout for home page according to the following sketch.



File Name : *hompage.dart*
Role : *Definition of Object Views as scaffold widget under HomePage class and their activities for Home page*

Significant Script:

- Imported packages:

```
import 'package:flutter/material.dart';
import 'package:flutter_face_recog2/Pages/test.dart';
import 'package:flutter_face_recog2/Pages/watch_stream.dart';
import 'face_recog.dart';
import 'face_registrer.dart';
```

- Edit Text-3:

```
  child:TextFormField(
controller:tEC3,
style: const TextStyle(
  fontSize: 18,
  color: Colors.black,
),
decoration: const InputDecoration(hintText: "IP Address",hintStyle:TextStyle(

  fontSize: 18,
  color: Colors.grey,
))),
```

- Button-2:

```
child: ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.deepPurpleAccent,
    elevation: 5,
// foreground
  ),
  onPressed: (){
      Navigator.push(context,
          MaterialPageRoute(builder: (context) =>
      WatchStreamPage(title: 'Watch Stream',ip: tEC3.text))
    );
  },
  child: const Text(
    'Watch Stream',
     style: TextStyle(
       fontWeight: FontWeight.w900,
       fontSize: 20,
       color: Colors.white,
     )
  ))),
```

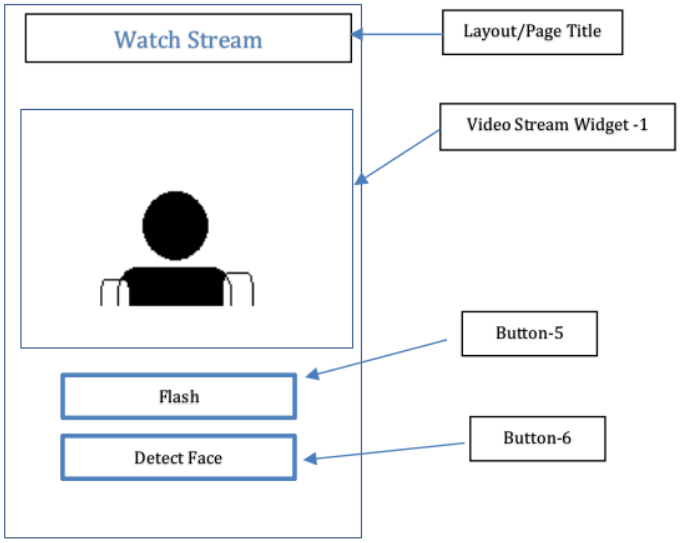
- Button-3:

```
child: ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.deepPurpleAccent,
          elevation: 5,
         // foreground
  ),
      onPressed: (){
        Navigator.push(context,
            MaterialPageRoute(builder: (context) =>
                FaceRegisterPage(title: 'Face Register',ip: tEC4.text))
        );
      },
      child: const Text(
        'Face Register',
         style: TextStyle(
           fontWeight: FontWeight.w900,
           fontSize: 20,
           color: Colors.white,
      ))),
```

- Button-4:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5, // foreground
    ),
    onPressed: (){
      Navigator.push(context,
          MaterialPageRoute(builder: (context) =>
              FaceRecognitionPage(title: 'Face Recognition',ip: tEC1.text))
      );
    },
    child: const Text(
      'Face Recognition',
       style: TextStyle(
         fontWeight: FontWeight.w900,
         fontSize: 20,
         color: Colors.white,
       )
    ))),
```

Step -4: Create a layout for Watch Video Stream page according to the following sketch.



File Name : *watch_stream.dart*
Role : *Definition of Object Views as scaffold widget under WatchStreamPage class and their activities for Watch Video Steam page*

Significant Script:

- Imported packages:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../vd.dart';
import 'package:flutter_mjpeg/flutter_mjpeg.dart';
import 'package:flutter_hooks/flutter_hooks.dart';
```

- Button-5:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,

      ), // foreground
      onPressed: () async {
        if(flash_on)
          {final response =await _connect.get("http://$ip/led?var=flash&val=0");
          flash_on=false;
          }
          else{
          await _connect.get("http://$ip/led?var=flash&val=1");
          flash_on=true;}
        _connect.get("http://$ip/stream");
      },
      child: const Text('Flash',
    style: TextStyle(
      fontWeight: FontWeight.w900,
      fontSize: 20,
      color: Colors.white,
    )))),
```

- Button-6:

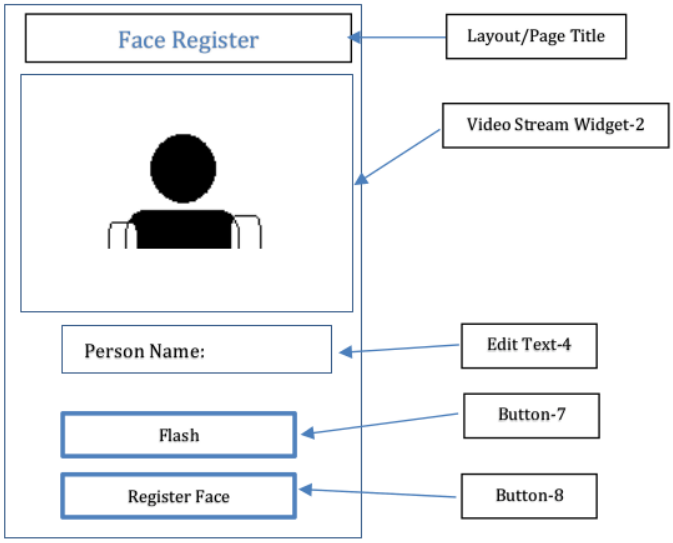
```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,

    ),

    onPressed: () async {
      if(detect_on)
      {final response =await _connect.get("http://$ip/detect?var=D1&val=0");
      detect_on=false;
      }
      else{
        await _connect.get("http://$ip/detect?var=D1&val=1");
        detect_on=true;}
    },
    child: const Text('Detect Face',
        style: TextStyle(
          fontWeight: FontWeight.w900,
          fontSize: 20,
          color: Colors.white,
        ))))
```

- Video Stream Widget -1:

```
Widget build(BuildContext context) {
  final isRunning = useState(true);
  _connect.get("http://$ip/recog?var=frecog&val=0");
  return Scaffold(
    appBar: AppBar(
      title: Text(title),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          SizedBox(
            width: 355,
            child: Mjpeg(
              isLive: isRunning.value,
              fit: BoxFit.fill,
              error: (context, error, stack) {
                print(error);
                print(stack);
                return Text(error.toString(),
                    style: const TextStyle(color: Colors.red));
              },
              stream: "http://$ip:81/stream",
            ),
          ),
```

.....

Step -5: Create a layout for Face Register page according to the following sketch.



File Name : *face_registrer.dart*
Role : *Definition of Object Views as scaffold widget under FaceRegisterPage class and their activities for Face Registration page*

Significant Script:

- Imported packages:

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_mjpeg/flutter_mjpeg.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:get/get_connect/connect.dart';
import 'package:video_player/video_player.dart';
import 'package:flutter_hooks/flutter_hooks.dart';
```

- Video Stream Widget -2:

```
Widget build(BuildContext context) {
  final isRunning = useState(true);
  _connect.get("http://$ip/recog?var=frecog&val=0");
  return Scaffold(
      appBar: AppBar(
        title: Text(title),
      ),
      body: Center(
        child: SingleChildScrollView(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              SizedBox(
                width: 355,
                child: Mjpeg(
                  isLive: isRunning.value,
                  fit: BoxFit.fill,
                  error: (context, error, stack) {
                    print(error);
                    print(stack);
                    return Text(error.toString(),
                        style: const TextStyle(color: Colors.red));
                  },
                  stream: "http://$ip:81/stream",
                ),
```

- Edit Text -4:

```
child: TextFormField(
  controller: tEC4,
  style: const TextStyle(
    fontSize: 18,
    color: Colors.black,
  ),
  decoration: const InputDecoration(
      hintText: "Person Name",
      hintStyle: TextStyle(
        //fontWeight: Colors.a,
        fontSize: 18,
        color: Colors.grey,
      )),
)),
```

- Button -7:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,
    ), // foreground
    onPressed: () async {
      if (flash_on) {
        final response = await _connect
            .get("http://$ip/led?var=flash&val=0");
        flash_on = false;
      } else {
        await _connect
            .get("http://$ip/led?var=flash&val=1");
        flash_on = true;
      }
    },
    child: const Text('Flash',
        style: TextStyle(
          fontWeight: FontWeight.w900,
          fontSize: 18,
          color: Colors.white,
        )))),
```

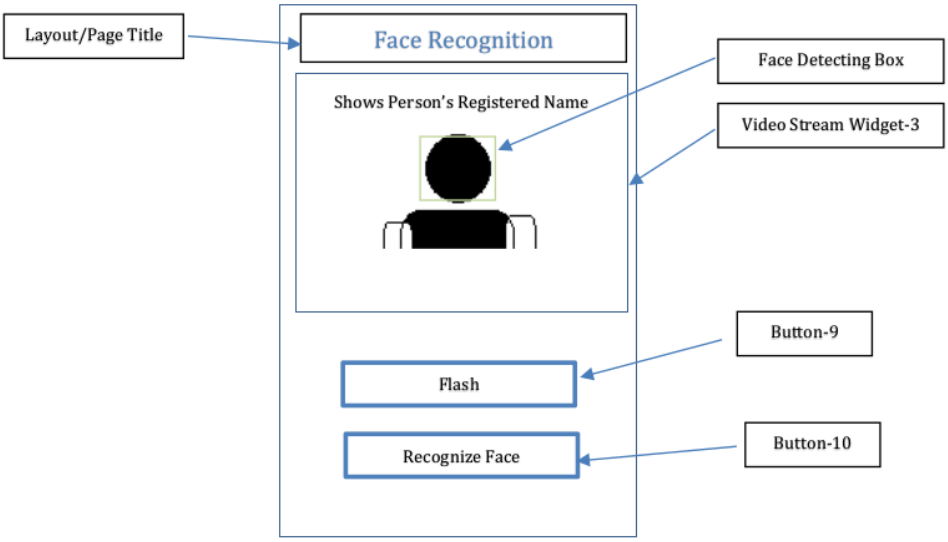
- Button -8:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,
    ),
    onPressed: () async {
      if (tEC1.text == "") {
        Fluttertoast.showToast(
            msg: "Input Name",
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            backgroundColor: const Color(0xFF82A30A),
            textColor: const Color(0xFF0D1141),
            fontSize: 16);
      } else {
        final response = await _connect.get(
            "http://$ip/enroll?var=${tEC1.text}&val=1");
        Timer.periodic(const Duration(seconds: 1),
            (timer) async {
          final resp = await _connect
              .get("http://$ip/rc?var=rc&val=1");
          String? a = resp.bodyString;
          a = a ?? "";
```

```
        if (a.contains("enc")) {
          Fluttertoast.showToast(
              msg: "Face Registration Completed",
              toastLength: Toast.LENGTH_LONG,
              gravity: ToastGravity.TOP,
              timeInSecForIosWeb: 1,
              backgroundColor: const Color(0xFF82A30A),
              textColor: const Color(0xFF0D1141),
              fontSize: 16);
          isRunning.value = !isRunning.value;
          timer.cancel();
        }
      });
    }
  },
  child: const Text('Register Face',
      style: TextStyle(
        fontWeight: FontWeight.w900,
        fontSize: 18,
        color: Colors.white,
      ))),
```

.....

Step -6: Create a layout for Face Recognition page according to the following sketch.



File Name : *face_recog.dart*
Role : *Definition of Object Views as scaffold widget under FaceRecognition class and their activities for Face Recognition page*

Significant Script:

- Imported packages:

```
import 'package:flutter/material.dart';
import 'package:flutter_mjpeg/flutter_mjpeg.dart';
import 'package:flutter_hooks/flutter_hooks.dart';
import 'package:video_player/video_player.dart';
import 'package:get/get.dart';
```

- Video Stream Widget -3:

```
Widget build(BuildContext context) {
  final isRunning = useState(true);
  _connect.get("http://$ip/recog?var=frecog&val=0");

  return Scaffold(
    appBar: AppBar(
      title: Text(title),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          SizedBox(
            width: 355,
            child: Mjpeg(
              isLive: isRunning.value,
              fit: BoxFit.fill,
              error: (context, error, stack) {
                print(error);
                print(stack);
                return Text(error.toString(),
                    style: const TextStyle(color: Colors.red));
              },
              stream: "http://$ip:81/stream",
            ),
          ),
```

- Button -9:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,
    ), // foreground
    onPressed: () async {
      if (flash_on) {
        final response = await _connect
            .get("http://$ip/led?var=flash&val=0");
        flash_on = false;
      } else {
        await _connect.get("http://$ip/led?var=flash&val=1");
        flash_on = true;
      }
    },
    child: const Text('Flash',
        style: TextStyle(
          fontWeight: FontWeight.w900,
          fontSize: 20,
          color: Colors.white,
        ))),
```

- Button -10:

```
child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.deepPurpleAccent,
      elevation: 5,
    ), // foreground
    onPressed: () async {
      if (recog_on) {
        final response = await _connect
            .get("http://$ip/recog?var=frecog&val=0");
        recog_on = false;
      } else {
        await _connect.get("http://$ip/recog?var=frecog&val=1");
        recog_on = true;
      }
    },
    child: const Text('Recognize Face',
        style: TextStyle(
          fontWeight: FontWeight.w900,
          fontSize: 20,
          color: Colors.white,
        ))))
```

Chapter 4

Results

After completion of all development and debugging tasks; the project should be verified, compiled and built in Android Studio and the outcome can be shown in three different ways.

1. Android Studio virtual Device (Emulator)
2. Flutter device Web Tools (Chrome/Edge)
3. A Real Android driven Smart Phone

4.1 Activating an Emulator

For convenience, the project results have been evaluated in an Android Studio Emulator and the AVD was created earlier having the following configurations.

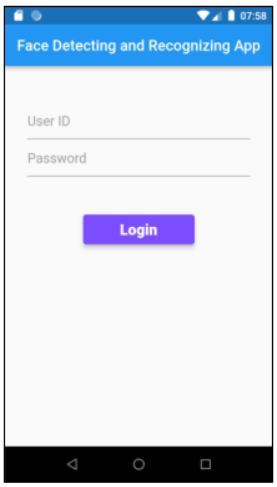
AVD Name	**:** Nexus S API 7
API Level	: 27
OS	: OREO Android 8.1 x86_64
Startup Orientation	: Portrait
Memory & Storage	: RAM – 1024 MB, VM Heap – 80 MB, Internal – 8000 MB, SD Card – Studio Managed,
Keyboard	: Enabled keyboard input

Before running the project, gradle files need to be checked for necessary inclusion of build libraries and exception handling. If any lacking exists in gradle file or flutter plugins or flutter plugin dependencies, it will be detected while compilation and to be fixed for successful run of the project.

4.2 App UI in Emulator

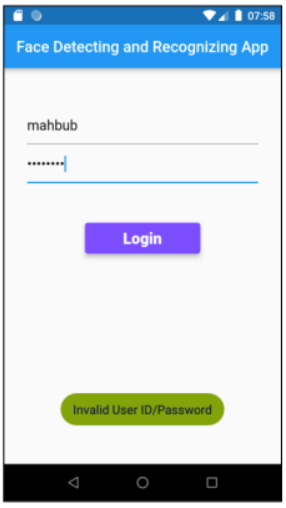
Final results are shown in next pages with the help of screen shot images taken from the AVD (Emulator).

Screenshot -1:



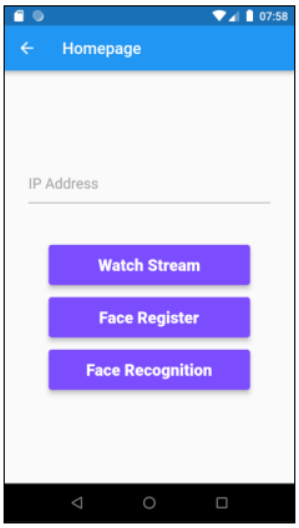
Activities : *First page of the app. Used to log in with predefined id-password.*

Screenshot -2:



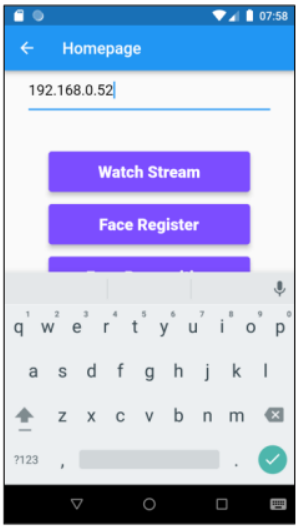
Activities : *Toast message alert for inputting wrong id or password.*

Screenshot -3:



Activities **:** *Showing the home page containing an edit text and three buttons. The meaning of hint text "IP Address" indicates the respective address of the ESP 32 Camera in Wi-fi LAN. It is possible to assign a static IP address in firmware codding for the camera. To increase portability of the camera and to avoid dependency to a fixed LAN, the IP address has kept editable for any network connectivity. To get the video stream of camera, Camera IP address needs to be inputted and then click 'Watch Stream' button.*

Screenshot -4:



Activities **:** Inputting IP address of the camera. *This is an essential parameter to input.*

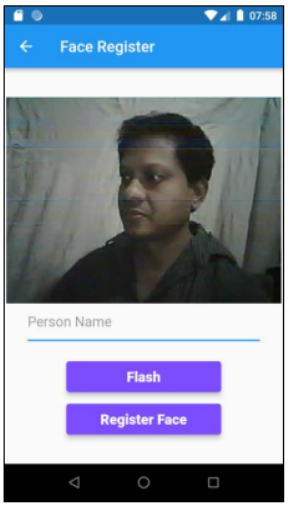
Screenshot -5:

Activities **:** *By clicking on 'Watch Stream' button in home page, this page will be found. The main function of this page is to show video stream captured by the ESP 32 camera. It works like a simple Closed-Circuit camera and conventional surveillance can be done through this option. There is a button to activate the 'Face detection' if necessary. The Flash button simply turn 'On' or 'Off' the onboard LED of ESP 32 Cam Board for using it in low light condition.*

Screenshot -6:

Activities **:** *The screen shot shows the effect of clicking on 'Detect Face' button.*

Screenshot -7:

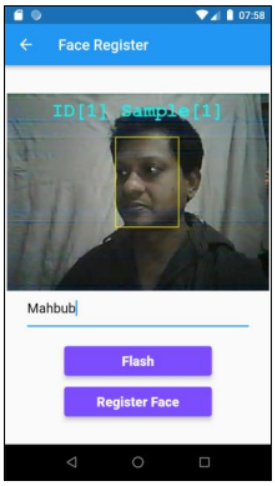


Activities **:** *After clicking on 'Face Register' button in home page, this page will appear. This is the most important part of face recognition mechanism. A person needs to enroll his face in front of the ESP 32 Camera to take several sample images and the system will store the sample images in memory. This process is calling here 'Face Registration' process. As like as previous page, flash button is also available here to increase lighting.*

When a person is present in front of Camera and 'Register Face' button gets clicked, the system first detect face and starts take samples in several attempts. In this project, the camera firmware is codded to take five samples for each face. So, it will take five sample images from different angles of the face from the video stream. After completion of taking samples a toast message in the app will be appeared to confirm user. Then it stops steaming of the camera captured video until further click on 'Register Face' button. The number of samples is a significant factor in face recognition system. The more samples are taken; more efficiently the system will work.

Face registration process is complex process. Extra care to be taken to ensure that maximum parts of the face are captured in image while taking samples. It is also important to avoid taking multiple samples from the same angle and of the same gesture. Adequate lighting is essential to take fine sample image. Proper placement of the camera and its steadiness must be ensured for capturing flicker free sample images. For using the camera in outdoor condition necessary sun shed and light reflection protection system should be arranged. It will require adequate water resistance arrangement too.

Screenshot -8:



Activities : *There is an Edit Text view in this page to input the person's name who is going to register his face in the system. It is mandatory to input a name in this field unless the 'Register Face' button will not work. After clicking on 'Register Face' button, the system immediately starts to take samples.*

Screenshot -9:

Activities : *Taking consecutive samples of the face to be registered by the system.*

Screenshot -10:

Activities : *Taking the sample from a different angle of the face to be registered.*

Screenshot -11:

Activities : *A toast message appears after taking five samples of face image which is the confirmation about completion of Face Registration process.*

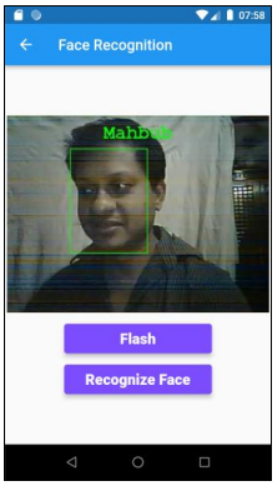
Screenshot -12:

Activities **:** *After getting back to Home Page and clicking on 'Face Recognition' button will bring the above page on screen. It will show the ultimate outputs of this project; that means the person's face that was registered earlier will be possible to recognize in video stream view of this page. To make this happened, 'Recognize Face' button needs to be clicked on and the Flash should be turned 'On' if lighting condition seems low for image quality. If the 'Recognize Face' is not clicked then it will just display the camera video.*

Screenshot -13:

Activities **:** *The system starts to recognize the face that was saved with a person name. The saved name is shown on the streaming video adjacent to the face recognizing box.*

Screenshot -14:



Activities : *The system is able to recognize the face from another angle and with a different gesture.*

Screenshot -15:

Activities : *The ESP 32 Camera captures a person's face that is not registered earlier in the system and the face is recognized as 'Unknown Face'.*

It is to be mentioned here that, a registered face can also be detected as unknown face while capturing the face from a different angle or having a gesture that is not registered in the system. Therefore, increasing the number of sample images registered from various angles and gestures will definitely increase the accuracy of recognizing efficiency.

Chapter 5

Conclusion

At the end in view, it may be concluded that the scope of this project has been compacted with its objectives i.e., to analyze a face recognition system with the help of ESP 32 CAM AI Thinker and visualize the face recognition results on an Android app using smartphones. The working procedure and results of this project render respective goals and envisioned for awarding the degree. The project may be evaluated on the basis of the following terms.

5.1 Discussion

The major characteristics of this project can be outlined as follows.

i) Simple architecture
ii) Introducing a modern development platform.
iii) Infinite Scalability
iv) Better implementation Feasibility
v) Low Cost

5.1.1 Simple architecture

The project was planned on the basis of a very simple architecture. To adopt the nature of educational project, this project used a methodology that is understandable to any engineering curriculum people of any language. The materialistic objective of this project is also to meet up one of our daily life problems like security and surveillance. The hardware and software components are very much known, available and very convenient to work with.

5.1.2 Introducing modern development platform

Smartphones are considered to be inseparable part of the latest technologies. There is no such device that is used frequently in our daily life. The development part of this project is done with the help of Android studio and its one of the latest development tools ‘Flutter’ which also gives a clear scenario of developing modern apps using powerful features of flutter.

5.1.3 Infinite Scalability

ESP 32 Cam board is a very popular Arduino board worldwide to the researchers. So many researchers worked with this camera interface to develop higher level applications of digital image processing. The AI thinker algorithm for detecting and recognizing the face of a person made this interface unparallel in comparison to other interfaces to build amazing projects. It can be programmed for real time simulation with a wide range of various purposes and suitable to integrate with the applications developed in any platform. This project may be considered as the 'kick start' of many sophisticated real time computer vision projects.

5.1.4 Better Implementation Feasibility

The features and operational procedure of this project have been kept so simple that the implementation of this project becomes feasible and able to meet up time constraints.

5.1.5 Low Cost

Required components of this project are available at a very low cost and its operational expenditure is zero. In comparison to other research projects, it was very cheaper to implement.

5.2 Other Relevant Projects

Most popular research projects and applications involve with Digital Attendance or HRM system, Criminal History, Image database Investigation, Payment Card Industries, Interactive Gaming, e-businesses by recognizing a customer and assessing the customer's needs.

5.3 Summary

The ultimate result of this project can be summarized as a face recognition system is developed which can monitor continuously to ensure surveillance of a restricted area. It has the ability to make an image database in memory card or a server by capturing different person's faces. And the system is capable to recognize the persons whose faces are stored in image database. The mobile application that is developed for the android smart phones have been tested in several devices and satisfactory result found. The accuracy level of face recognition performance of the system was fixed for educational purpose. Accuracy level should be increased for real life applications.

5.4 Recommendations

In this project, the image database is maintained in micro-SD card or internal memory of the ESP 32 Cam board which is limited to store maximum 4 GB data. Besides, the camera firmware is codded to take only five image samples for each face. To increase samples per face the system needs more storage and processing power. Is this case, OpenCV (Open-Source Computer Vision Library) can be applied and user interface can be programmed using the 'Python' language. It will help to develop robust face recognition system. In this environment, sample image database is maintained in a server with larger storage and huge number of sample images per face can be stored. That's why it can recognize faces at an outstanding level.

References

[1] A. Kaur, A. Jadil, A. Sadhu, S. Goyal, A. Mehra, M. Rahul, "Cloud Based Survellance using ESP32 CAM", International Conference on Intelligent Technology, System and Service for Internet of Everything, 2021.

[2] N. H. Barnouti, S.S. Mahmood, W.E. Matti, "Face Recognition: A Literature Review," in International Journal of Applied Information Systems (IJAIS)-ISSN:2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 11 No. 4, September 2016 – www.IJAIS.org.

[3] Nicholl, P. and Amira, A., 2008. "DWT/PCA face recognition using automatic coefficient selection". In Electronic Design, Test and Applications", 2008. 4th IEEE International Symposium on, pp. 390-393. IEEE

[4] Kshirsagar, V.P., Bavisskar, M.R. and Gaikwad, M.E., 2011, March. "Face Recognition using Eigenfaces". In Computer Research and Development (ICCRD), 2011, 3rd International Conference on, pp. 302-306. IEEE.

[5] Pornpanomchai, C. and Inkuna, C., "Human face recognition by Euclidean distance and neural network"., In Second International Conference on Digital Image Processing. International Society for Optics and Photonics. 2014, pp. 754603-754603.

[6] Chellappa, R., Wilson,C.L. and Sirohey, S., "Human and machine recognition of faces: A survey". 1995, Proceeding of the IEEE, 83 (5), pp. 705-741.

[7] Kadam, K.D., "Face Recognition using Principal Component Analysis with DCT". 2014, International Journal of Engineering Research and General Science, ISSN, pp. 2091-2730.

[8] Sharma, N. and Dubey, "Face Recognition Analysis using PCA, ICA, and Neural Network". International Journal of Digital Application & Contemporary research, 2(9), 2014.

[9] Nigam, A. and Gupta, P., "A new distance measure for face recognition system". In Image and Graphics, 2009, ICIG. Fifth International Conference on, pp. 696-701. IEEE.

d

[10] Kasar, M.M. Bhattacharyya, D. and Kim, T.H., "Face Recognition Using Neural Network: A Review". 2016. International Journal of Security and Its Applications, 10(3), pp. 81-100.

[11] Bakshi, Y., Kaur, S. and Verma, P., "An Improvement in Face Recognition for Invariant faces. 6(2), 2016. pp. 423-426,

[12] Poon, B., Amin, M.A. and Yan, H., "Improved Methods on PCA Based Human Face Recognition for Distorted Images". 2016. In proceedings of the International Multi Conference of Engineers and Computer Scientists (1).

[13] Singh, N.A., Kumar, M.B. and Bala, M.C., "Face Recognition System based on SURF and LDA Technique". 2016. International Journal of Intelligent Systems and Applications, 8(2).

[14] Bhat, V.S. and Pujari, J.D., "Human Face Recognition Using Combined Approaches PCA and ICA". 2015. Digital Image Processing, 7(10), pp. 301-307.

[15] Vinay, A., Shekhar, V.S., Murthy, K.B. and Natarajan, S., "Face Recognition Using Gabor Wavelet Features with PCA and KPCA - A Comparative Study". 2015. Procedia Computer Science, 57, pp. 650-659.

[16] Taghizadegan, Y., Ghassemian, H. and Naser-Moghaddasi, M., "3D Face Recognition Method Using 2DPCA-Euclidean Distance Classification". 2012. *ACEEE International Journal on Control System and Instrumentation, 3(1)*.

[17] Wang, L., Zhang, Y. and Feng, J., 2005. On the Euclidean distance of images. IEEE transactions on pattern analysis and machine intelligence, 27(8), pp.1334-1339.

[18] Banzi, M., Cuartielles, D., Igoe, T., Mellis, D., “Arduino PRO: Edge IoT technology”, Arduino Software, July’2022. https://forum.arduino.cc/t/spi-uno-r3-and-esp32-cam-ai-thinker/1017219/12

[19] Banzi, M., Cuartielles, D., Igoe, T., Mellis, D., “Arduino PRO: Edge IoT technology”, Arduino Software, July’2022. https://forum.arduino.cc/t/spi-uno-r3-and-esp32-cam-ai-thinker/1017219/12.

[20] Shanthar, P., How to Electronics, “ESP32 CAM Based Face & Eye Recognition System”, August 20, 2022.https://how2electronics.com/esp32-cam-based-face-eyes-recognition-system/

[21] Santos, R. and Santos S., “ESP 32 CAM Video Streaming and Face Recognition with Arduino IDE | random nerd tutorials”, https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

[22] Imf, “Build Your First App in Java | Android Developers”, January 8, 2021. https://developer.android.com/codelabs/build-your-first-android-app#0

[23] “User Interface | Flutter”. https://docs.flutter.dev.development /ui

[24] “Creating Simple Application in Android Studio”. https://www.tutorialspoint.com/flutter/flutter_creating_simple_application_in_android_studio.htm#

- The End -

Project_Report_draft_v1.6.docx

ORIGINALITY REPORT

13%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.ijais.org Internet	395 words — 6%
2	www.slideshare.net Internet	91 words — 1%
3	www.coursehero.com Internet	89 words — 1%
4	sites.google.com Internet	32 words — < 1%
5	"3D Shape Analysis", Wiley, 2019 Crossref	28 words — < 1%
6	Alexander Dubelschikov, Tatyana Tsoy, Yang Bai, Mikhail Svinin, Evgeni Magid. "Intelligent System Concept of an IoT Cameras Network Application for an Unmanned Aerial Vehicle Control via a Graphical User Interface", 2022 International Conference on Information, Control, and Communication Technologies (ICCT), 2022 Crossref	26 words — < 1%
7	www.ntu.edu.sg Internet	23 words — < 1%
8	Lecture Notes in Computer Science, 2007. Crossref	19 words — < 1%

9	www.jgrcs.info Internet	18 words — < 1%
10	Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering, 2013. Crossref	16 words — < 1%
11	"Biometric Recognition", Springer Science and Business Media LLC, 2018 Crossref	14 words — < 1%
12	Bintang Wahyudono, Dion Ogi. "Implementation of Two Factor Authentication based on RFID and Face Recognition using LBP Algorithm on Access Control System", 2020 International Conference on ICT for Smart Society (ICISS), 2020 Crossref	14 words — < 1%
13	eajse.org Internet	14 words — < 1%
14	www.tutorialspoint.com Internet	12 words — < 1%
15	Han, Peng Wu, Cong Ji, Si Ming Wang, Dai Xing Zhang, and Dong Yang. "Design of an Authentication System Based on Face Recognition and the Second Generation ID Detection", Advanced Materials Research, 2012. Crossref	11 words — < 1%
16	github.com Internet	11 words — < 1%
17	prr.hec.gov.pk Internet	11 words — < 1%

18	how2electronics.com Internet	10 words — < 1%
19	www.amrita.edu Internet	10 words — < 1%
20	bbrc.in Internet	9 words — < 1%
21	eprints.utm.my Internet	9 words — < 1%
22	ijeei.org Internet	9 words — < 1%
23	J. Subhashini, Aayush Sharma, Budhaditya Dutta, Varnit Vijay, Shivam Verma. "Voice-controlled movable smart dustbin", AIP Publishing, 2022 Crossref	8 words — < 1%
24	cueltconferences.org Internet	8 words — < 1%
25	hdl.handle.net Internet	8 words — < 1%
26	M S U Sarwar, A Sharin, M R Khan, H Imtiaz, S A Fattah. "A face recognition scheme based on spatial correlation function", TENCON 2010 - 2010 IEEE Region 10 Conference, 2010 Crossref	6 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	OFF	EXCLUDE MATCHES	OFF