

FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON WITH ESP32

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

by

V. ROFIN SHALOM (40130197)

S. SAJEEV (40130200)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

ACCREDITED WITH GRADE "A++" by NAAC

CATEGORY – 1 UNIVERSITY BY UGC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2024



SATHYABAMA

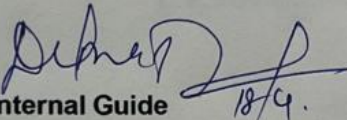
**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **V.ROFIN SHALOM (40130197), S. SAJEEV (40130200)** who carried out the project entitled "**PROJECT FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON WITH ESP32 CAM**" under our supervision from November 2023 to April 2024.


Internal Guide

Dr. T. THAJ MARY DELSY, M.E., Ph.D.,



Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **V.Rofin Shalom (40130197)** and **S.Sajeev (40130200)** hereby declare that the Project Report entitled **“FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON WITH ESP32”** done by us under the guidance of **“Dr.T.Thaj Mary Delsy,M.E.,Ph.D.,”** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering/ Technology degree in **Electronics and Communication Engineering.**

DATE:18\04\2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

1:

2:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and successfully completing it. I am grateful to them.

I convey my thanks to Dr. **N. M. NANDHITHA, M.E.,Ph.D.**, Dean, School of Electrical and Electronics Engineering, and **Dr.T. RAVI, M.E.,Ph.D.**, Head of the department, Department of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep gratitude to our project guide **Dr.T.Thaj Mary Delsy, M.E., Ph.D.**, for her valuable guidance, suggestions, and constant encouragement that paved the way for our project's successful completion work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Face recognition is one of the most useful applications for image processing and plays a crucial role in the technical field. For authentication purposes, specifically in the context of student attendance, face recognition is a pressing issue. The process of recognizing students using face biostatistics based on high-definition monitoring and other computer technologies is known as an attendance system using face recognition. The goal of developing this system is to digitize the old method of taking attendance by calling names and keeping pen-and-paper records. The current methods for taking attendance are time-consuming and arduous.

Manual recording makes it simple to alter attendance records. Both the current biometric systems and the conventional method for taking attendance are susceptible to proxies. As a result, it is suggested that this paper address all of these issues. Haar classifiers, KNN, CNN, SVM, generative adversarial networks, and Gabor filters are utilized in the proposed system. Attendance reports will be created and saved in Excel format following face recognition. The system is put through its paces under a variety of conditions, including lighting, head movements, and varying distances between the student and the cameras. The overall complexity and accuracy are determined after extensive testing. The proposed system proved to be a time- and labor-saving and durable tool for taking attendance in a classroom. The developed system is cost-effective and requires less installation.

INDEX

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
3	AIM AND SCOPE	8
	3.1 AIM	8
	3.2 SCOPE	8
	3.3 PROPOSED SYSTEM	8
	3.3.1 ADVANTAGES	9
	3.3.2 DISADVANTAGES	9
4	METHODS, METHODOLOGY AND MATERIAL USED	10
	4.1 BLOCK DIAGRAM FOR IMAGE PROCESSING	10
	4.2 IMAGE PROCESSING	10
	4.2.1 IMAGE ACQUISITION	10
	4.2.2 IMAGE ENHANCEMENT	11
	4.2.3 IMAGE RESTORATION	11
	4.2.4 COLOUR IMAGE PROCESSING	11
	4.2.5 WAVELETS	11

	4.2.6 COMPRESSIONS	11
	4.2.7 MORPHOLOGICAL PROCESSING	11
	4.2.8 SEGMENTATION	11
	4.2.9 REPRESENTATION AND DESCRIPTION	11
	4.2.10 OBJECT RECOGNITION	11
	4.3 CIRCUIT DIAGRAM	12
	4.4 FLOW CHART	13
	4.5 HARDWARE REQUIREMENT	14
	4.5.1 RECTIFIER	14
	4.5.2 TRANSFORMER	14
	4.5.3 ESP-32 CAM	15
	4.5.4 CAPACITOR	15
	4.5.5 DIODE	16
	4.6 SOFTWARE REQUIREMENT	17
	4.6.1 PYTHON	17
	4.6.2 ARDUINO IDE	18
	4.6.3 EMBEDDED C	19
5	RESULTS AND PERFORMANCE ANALYSIS	20
	5.1 PROGRAM CODE	20
	5.2 WORKING OF PROJECT	22
	5.3 OUTPUT	26
6	CONCLUSION	27
	6.1 RESULT	27

	6.2 SUMMARY	28
	6.3 FUTURE WORK	28
7	REFERENCES	29

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	IMAGE PROCESSING	10
4.2	CIRCUIT DIAGRAM	12
4.2	FLOW CHART	13
5.1	PYTHON CODE	22
5.2	VISUAL STUDIO	23
5.3	ZIP FOLDER	24
5.4	IMAGE FOLDER	24
5.5	FACE DETECTION	25
5.6	OUTPUT	26

Chapter-1

INTRODUCTION

We address the problem of 3-D face tracking, which is a fundamental problem in the context of human computer interaction. The input consists in a video stream acquired by a low-cost webcam pointing to the user's face. Our efforts focus on real-time and robust full 3-D face tracking (6 DOF), in the presence of facial expression changes, wide pose variations, and partial occlusions. To this end, we make very few assumptions compared to most of the state-of-the-art approaches. The distance to the camera should comply with the requirements of a facial interactive application, but can vary according to the focal length used. Also, we require a near frontal pose at initialization. The proposed approach includes the use of a generic 3-D face model to track in real-time the 3-D head motion of users and produce accurate and textured face models. As illustrated in Fig. 1, our application is composed of three main modules: initial 3-D model fitting, 3-D head tracking, and re-acquisition. The applications of the 3-D tracker and 3-D model are not discussed here. We need a 3-D model, which can either be a generic model, or a specific one retrieved from a database. In the initial 3-D modeling step, the model is warped orthogonally to the focal axis in order to fit to the user's face in an input image, by matching 2-D facial landmarks extracted at run time. The initial 3-D reconstruction of the face model is a critical step.

Face detection is a computer technology that is being applied for many different applications that require the identification of human faces in digital images or video. It can be regarded as a specific case of object-class detection, where the task is to find the locations and sizes of all objects in an image that belongs to a given class. The technology is able to detect frontal or near-frontal faces in a photo, regardless of orientation, lighting conditions or skin colour.

OpenCV comes with a trainer as well as a detector. If you want to train your own classifier for any object like a car, planes, etc. you can use OpenCV to create one. Its full details are given here: [Cascade Classifier Training](#).

Here we will deal with detection. OpenCV already contains many pre-trained classifiers for face, eyes, smile, etc. Those XML files are stored in OpenCV/data/haarcascades/ Folder. Let's create a face detector with OpenCV.

The growing interest in computer vision of the past decade. Fueled by the steady doubling rate of computing power every 13 months, face detection and recognition has transcended from an esoteric to a popular area of research in computer vision and one of the better and successful applications of image analysis and algorithm-based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuro-scientific and psychological studies, mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa. Because of general curiosity and interest in the matter, the author has proposed to create an application that would allow user access to a particular machine based on an in-depth analysis of a person's facial features. This application will be developed using Intel's open-source computer vision project, OpenCV

Face recognition is an easy task for humans. Experiments in [6] have shown, that even one-to-three-day old babies are able to distinguish between known faces. So how hard could it be for a computer? It turns out we know little about human recognition to date. Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? How do we analyse an image and how does the brain encode it? It was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them. Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. One of the first automated face recognition systems was described in marker points (position of eyes, ears, nose,) were used to build a feature vector (distance between the points, angle between them, ...). The recognition was performed by calculating the Euclidean distance between feature vectors of a probe and reference image. Such a method is robust against changes in illumination by its nature, but has a huge drawback: the accurate registration of the marker points is complicated, even with state-of-the-art algorithms. Some of the latest work on geometric face recognition was carried out in. A 22-dimensional feature vector was

used and experiments on large datasets have shown, that geometrical features alone don't carry enough information for face recognition. The Eigen faces method described in took a holistic approach to face recognition: A facial image is a point from a high-dimensional image space and a lower-dimensional representation is found, where classification becomes easy. The lower-dimensional subspace is found with Principal Component Analysis, which identifies the axes with maximum variance. While this kind of transformation is optimal from a reconstruction standpoint, it doesn't take any class labels into account. Imagine situation where the variance is generated from external sources, let it be light. The axes with maximum variance do not necessarily contain any discriminative information at all, hence a classification becomes impossible. So, a class-specific projection with a Linear Discriminant Analysis was applied to face recognition.

Chapter-2

LITERATURE SURVEY

Chadi Bozazi et.al., in 2022, of an IOT System based on Face Recognition Technology using ESP32-CAM. In this paper, we will present the realization of a facial recognition system using the ESP32-CAM board controlled by an Arduino board. The goal is to monitor a remote location in real time via a camera that is integrated into the ESP32 IOT board. The acquired images will be recorded on a memory card and at the same time transmitted to a pc (a web server). The development of this remote monitoring system is to create an alternative between security, reception, and transmission of information to act accordingly. The simulation results of our proposed application of the facial recognition domain are very efficient and satisfying in real time

Adrian Barbu et.al., in 2019 This paper presents a part-based face detection approach where the spatial relationship between the face parts is represented by a hidden 3D model with six parameters. The computational complexity of the search in the six dimensional pose space is addressed by proposing meaningful 3D pose candidates by image-based regression from detected face key point locations. The 3D pose candidates are evaluated using a parameter sensitive classifier based on difference features relative to the 3D pose. A compatible subset of candidates is then obtained by non-maximal suppression. Experiments on two standard face detection datasets show that the proposed 3D model based approach obtains results comparable to or better than state of the art. Face recognition has been a hot research area for its wide range of applications . In human identification scenarios, facial metrics are more naturally accessible than many other biometrics, such as iris, fingerprint, and palm print . Face recognition is also highly valuable in human computer interaction, access control, video surveillance, and many other applications.

Although 2D face recognition research made significant progresses in recent years, its accuracy is still highly depended on light conditions and human poses . When the light is dim or the face poses are not properly aligned in the camera view, the recognition accuracy will suffer.

Jongmoo Choi et.al., in 2018 , We first infer a 3-D face model from a single frontal image using automatically extracted 2-D landmarks and deforming a generic 3-D model. Then, for any input image, we extract feature points and track them in 2-D. Given these correspondences, sometimes noisy and incorrect, we robustly estimate the 3-D head pose using PnP and a RANSAC process. As the head moves, we dynamically add new feature points to handle a large range of poses.

This article objective is to implement an eye-blink detection-based face likeness detection algorithm to thwart photo attacks. The algorithm works in real time through a webcam and displays the person's name only if they blinked. In layman's terms, the program runs as follows: Detect faces in each frame generated by the webcam.

For each detected eyes, detect if eyes are open or closed If at some point it was detected that the eyes were open then closed then open, we conclude the person has blinked and the program displays its name (in the case of a facial recognition door opener, we would authorize the person to enter).

Sneha Medhava et.al., Attendance recording is critical in current environment for increasing the quality of any industrial system. Accordingly, a strong computerised Face detection attendance recording system using an ESP32 camera is required. Every organisation has a stringent policy requiring attendance. Maintaining an attendance register manually involves effort and time. RFID and biometric technology, for instance, are two approaches to the same problem. This analysis describes an effective and inventive technique for validating attendance and detecting presence. The ESP32 microcontroller family has dual-mode Bluetooth and built-in Wi-Fi. In contrast to biometric attendance systems, this facial detectionbased technology does not require someone to be close to the fingerprint reader module. When the same student or employee uses the biometric attendance system to record their attendance, the biometric device becomes contaminated. And in case of RFID chances of proxy is high. Using facial recognition technology, this forecast will track employee or student attendance.

N.Bayramoglu, G. Zhao et.al.,2017.A real-time, GUI based automatic Face detection and recognition system is developed in this project. It can be used as

access control system by registering the staff or students of an organization with their faces, and later it will recognize the people by capturing their images with faces, when they are entering or leaving the premises. The system is implemented on a desktop with a Graphical User Interface, Initially it detects the faces in the images that are grabbed from a web camera. All the tools and operating, used to develop this system like Ubuntu, open Face, Python ..., are open source tools. This real time GUIbasedface recognition system is developed using Open source tool Open face. Open Face is the face recognition tool developed by Carnegie Mellon University, using OpenCV. Open Face, consists in a broader Prospective, three phases: Detection, Feature extraction, and Recognition. The dimensionality of face image is reduced by the Histogram of Oriented Gradients (HOG) and this algorithm is developed to detect frontal views of faces. After detecting the face part of image, extract the 128 face features for the given image by using a Deep Neural Network algorithm and the recognition is done by the Support Vector machine (SVM) classifier.HOG is one of the most popular representation methods for a face image. It not only reducesthe dimensionality of the image, but also extracting the facial features of the given images, and retains some of the variations in the image data. So dimensionality of face image reduced by HOG using deep learning algorithm and recognition is done by SVM approach

Okaty Turkdagli et.al.,in 2023 Authentication is the confirmation of the accuracy of the data piece that any institution, person or system accepts as correct. Many methods are used for the authentication process. Some of these are the methods using biometric data such as face authentication, fingerprint authentication, and iris authentication. In this article, the way to create a secure system using face recognition and sign language as an authentication method is discussed and an application using ESP32-CAM is developed and tested. The results show that secure authentication cannot be achieved with facial recognition and sign language. The developed system is low cost and easy to implement. With this system, authentication can be done without requiring any physical contact, and it can be used for personal security and entrance and exit. Sign language, which is frequently used by hearing-impaired individuals, can play an active role as authentication. With low-cost modules such as ESP32, it will be an alternative to authentication in almost every environment.

Jamil Abedalrahim et.al., in 2022 Face recognition technology is used in biometric security systems to identify a person digitally before granting the access to the system or the data in it. There are many kidnappings or abduction cases happen around us, however, the kidnap suspects will be set free if there is lack of evidence or when the victims are not able to testify in court because they suffer from post-traumatic stress disorder (PTSD). The objectives of this study are, to develop a device that will capture the image of a kidnapper as evidence for future reference and send the captured image to the family of the victim through email, to design a face recognition system to be used in searching kidnap suspects and to determine the best training parameters for the convolution neural network (CNN) layers used by the proposed face recognition system. The accuracy of the proposed system is tested with three different datasets, namely the AT&T database, face database from [23] and a custom face dataset.

CHAPTER-3

AIM AND SCOPE

3.1 Aim

The main aim of the face recognition using ESP32 can also allow detecting a face. All you have to try multiple attempts of enrolling your face with a different angle. Once it is enrolled, it will detect you and someone not enrolled in its database when you come in front of the camera. It will give you an intruder alert. The aim and scope of face recognition using an ESP32-CAM module typically revolve around creating a low-cost, embedded system capable of detecting and recognizing human faces.

3.2 SCOPE

The suggested system's job is to take a picture of every student or employee's face and record it in a database for their attendance. All of the student's or employee's facial features, as well as the student or employee's seating position and posture, must be recorded while the face of the subject is being photographed. The instructor does not have to manually take attendance. Because the system records a video of the class, and after further processing, the face is identified and the attendance database is updated. The technology utilizes the automatic recognition of attendance that is achieved through ongoing monitoring. The performance of the attendance can be estimated and improved with constant observation. Students present in the classroom are photographed from their postures and faces in order to record their attendance. The technology calculates each student's seating arrangement and location for use in keeping track of attendance through constant observation and recording. The way to determine the various weights for each focused seat according to its placement is the main focus of the work. To enable quicker recognition of the image, the effectiveness of the picture is also being discussed.

3.3 PROPOSED SYSTEM

This proposed system is mainly about the face detection-based attendance system using the image processing technique. And here we using the face recognition system with esp32 camera module. And the main task is to take a

picture of every student or employee's face and record it in a database for their attendance in excel. And check the employee/student sample with the existing database and verify whether that he/she is existing person or intruder.

3.3.1 ADVANTAGES

- Here we use ESP32 Cam for verification.
- Daily Attendance will be noted in the excel sheet.
- Implementation is easy.
- Cost is economical.
- Efficiency is high compared to existing system

3.3.2 DISADVANTAGES

- Here the implementation takes much time.
- Accuracy value is too low
- Face recognition latency is high.
- Attendance won't get registered in the excel sheet.

CHAPTER – 4

METHODS, METHODOLOGY AND MATERIALS USED

4.1 Block Diagram For Image Processing

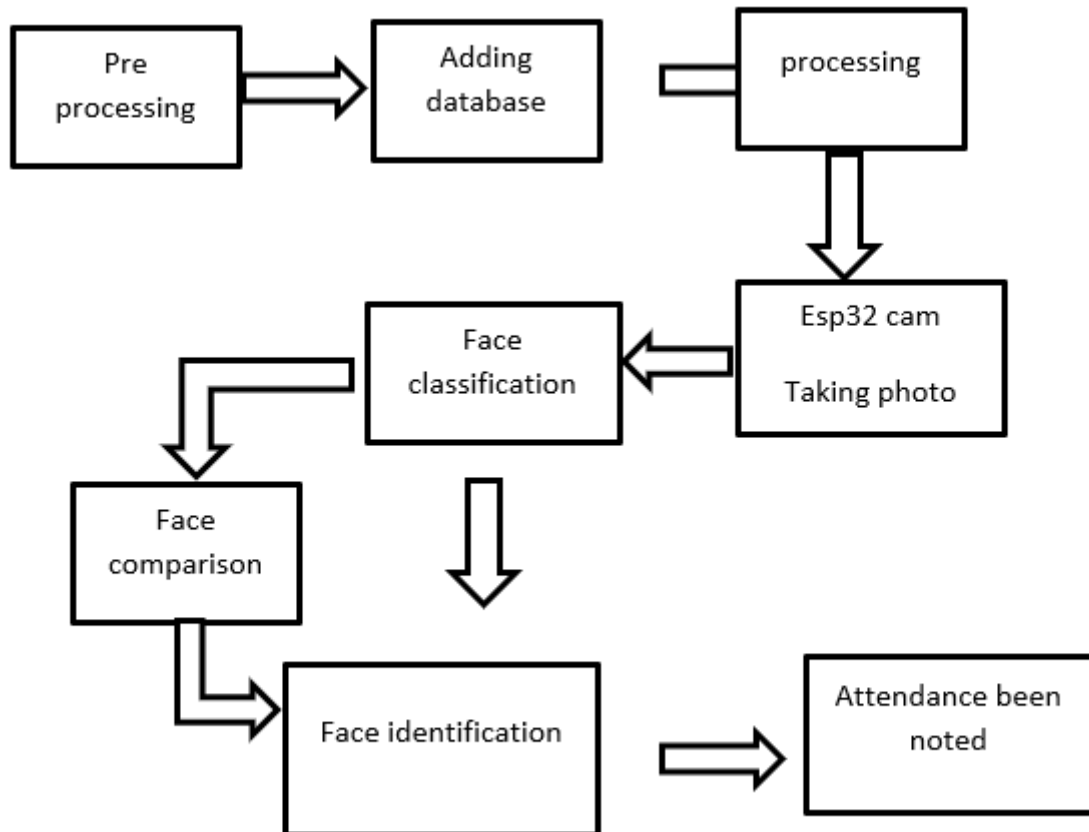


Fig 4.1 For image processing

4.2 IMAGE PROCESSING EXPLANATION

4.2.1 Image Acquisition

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling, etc.

4.2.2 Image Enhancement

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as changing brightness & contrast, etc.

4.2.3 Image Restoration

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

4.2.4 Color Image Processing

Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

4.2.5 Wavelets and Multiresolution Processing

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

4.2.6 Compression

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of the internet, it is very much necessary to compress data.

4.2.7 Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

4.2.8 Segmentation

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward a successful solution of imaging problems that require objects to be identified individually.

4.2.9 Representation and Description

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing.

4.2.10 Object recognition

Recognition is the process that assigns a label, such as, vehicle to an object based on its descriptors.

4.2.11 Knowledge Base

Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information.

The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

4.3 CIRCUIT DIAGRAM

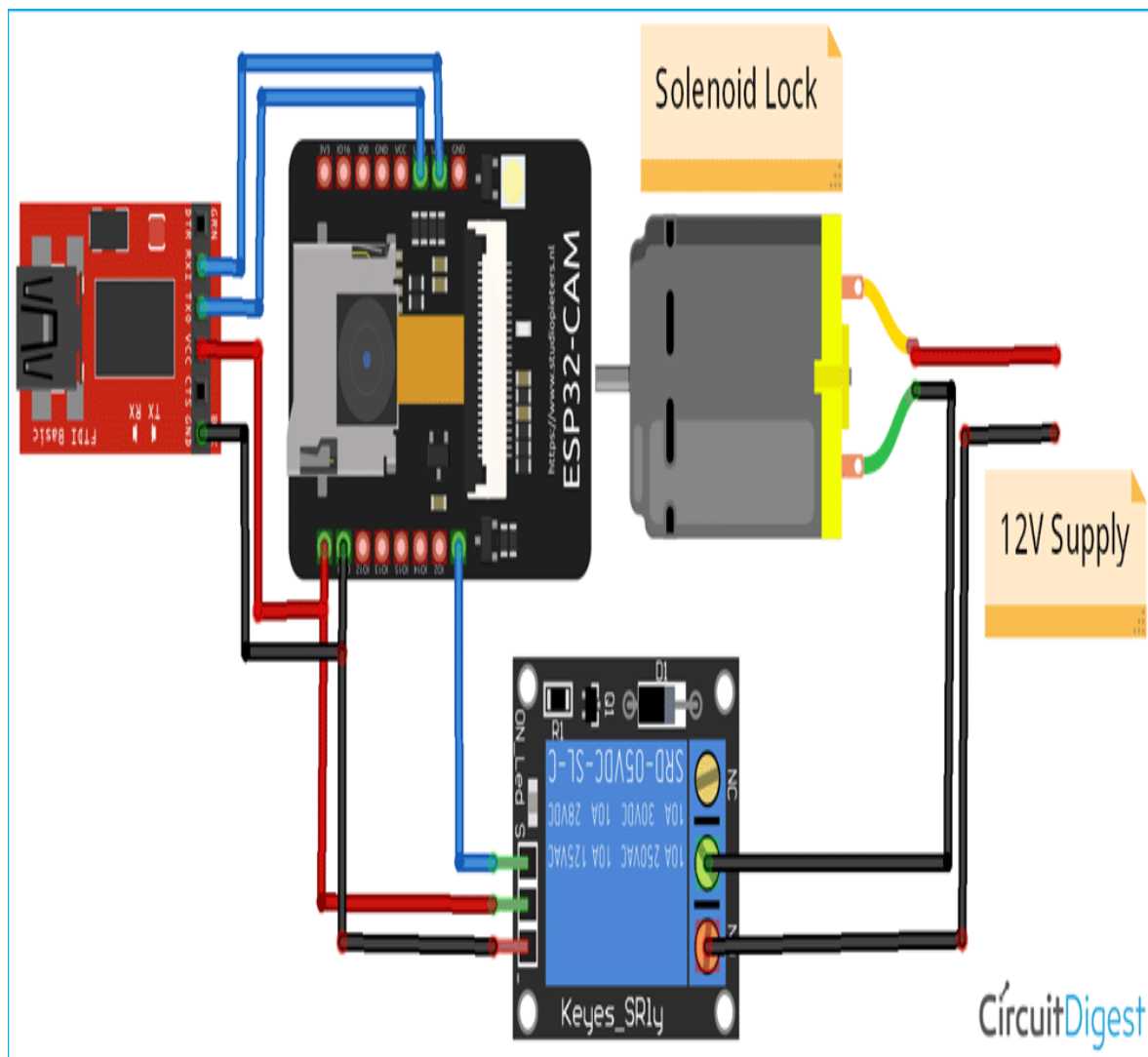


Fig 4.2 CIRCUIT DIAGRAM

4.4 FLOW CHART

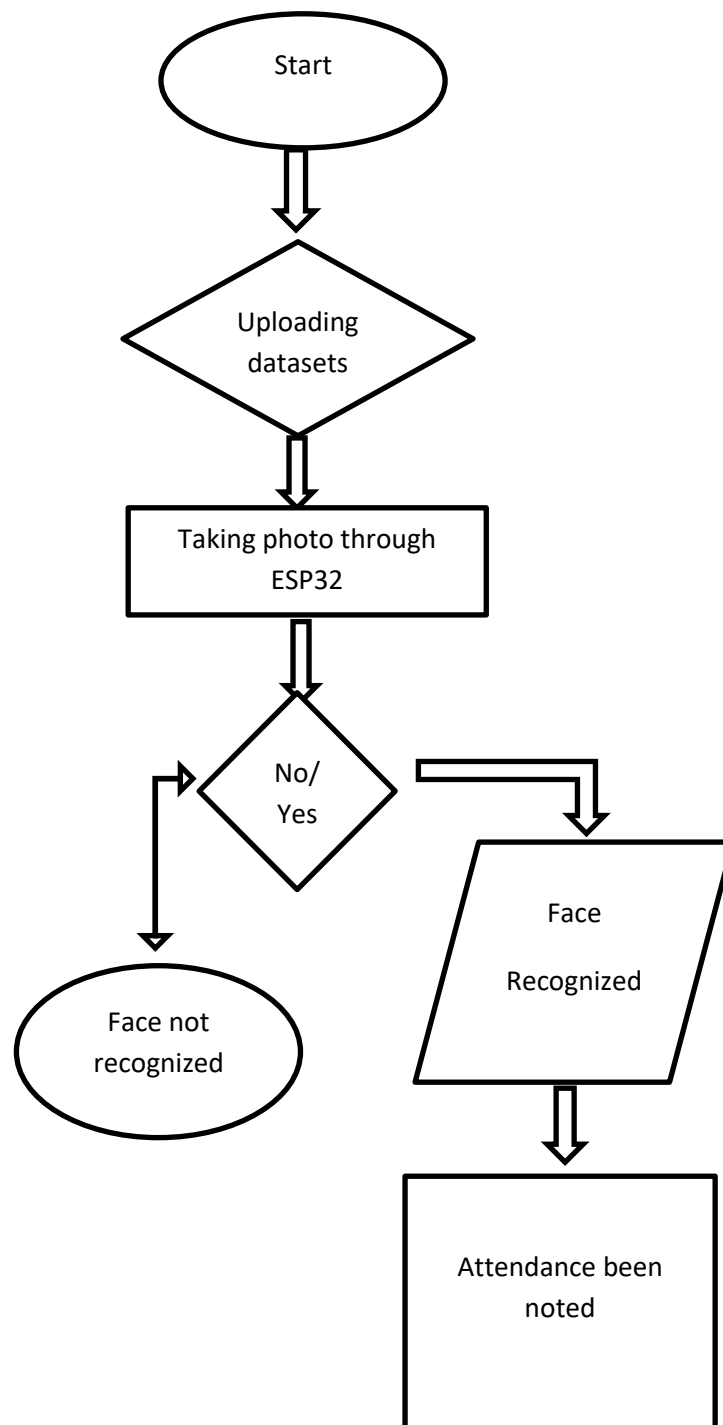


Fig:4.3 FLOW CHART

4.5 HARDWARE REQUIREMENTS

4.5.1 RECTIFIER

A rectifier is an electronic device that converts an alternating current into a direct current by using one or more P-N junction diodes. A diode behaves as a one-way valve that allows current to flow in a single direction. This process is known as rectification. A full wave rectifier is defined as a rectifier that converts the complete cycle of alternating current into pulsating DC. Unlike halfwave rectifiers that utilize only the halfwave of the input AC cycle, full wave rectifiers utilize the full cycle. There are two main types of rectifiers— uncontrolled and controlled. Controlled rectifiers: These are the types of rectifiers whose voltage can be changed. To convert an uncontrolled rectifier into a controlled rectifier, MOSFETs, IGBTs, or SCRs are used. These are more desirable than their uncontrolled counterparts. A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes (say D2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period $t = 0$ to $T/2$. Accordingly for the negative cycle of the input the conducting diodes are D1 & D4. Thus the polarity across the load is the same. In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, 1N4007 etc.... can be used. But here we use 1N4007, because it can withstand up to 1000v.

4.5.2 TRANSFORMER

A Step-down transformer decreases the voltage incoming to the site by increasing the electrical current. It does this by converting the high incoming voltage in the primary winding to the necessary lower voltage in the secondary windings. step-down transformer is an electrical device that reduces the voltage of an alternating current (AC) power supply. It consists of a primary winding, a secondary winding, and an iron core.

A transformer is a static piece of which electric power in one circuit is transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our

project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac. When an AC voltage is applied to the primary winding, it creates a fluctuating magnetic field in the iron core. This magnetic field then induces a voltage in the secondary winding but at a lower voltage level than the primary winding. A transformer works on the mutual induction principle, also known as Faraday's Law of Electromagnetic Induction, which states that the magnitude of voltage is directly proportional to the rate of change of magnetic flux.

To get a deeper view of the performance of a transformer, let's understand its basics.

4.5.3 ESP32-CAM

The ESP32-CAM is a very small camera module with the ESP32-S chip that costs approximately \$10. Besides the OV2640 camera, and several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the camera or to store files to serve to clients. Here are the features of ESP32CAM. The smallest 802.11b/g/n Wi-Fi BT SoC module.

- Low power 32-bit CPU, can also serve the application processor
- Up to 160MHz clock speed, summary computing power up to 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, built-in flash lamp
- Support image WiFi upload
- Support TF card
- Supports multiple sleep modes
- Embedded Lwip and FreeRTOS
- Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

4.5.4 CAPACITOR

A **capacitor** is a device in which electrical energy can be stored. It is an arrangement of two conductors, generally carrying charges of equal magnitudes and

opposite signs, and separated by an insulating medium. The non-conductive region can either be an electric insulator or vacuum, such as glass, paper or air, or a semiconductor called a dielectric. Capacitors vary in shape and size, and they have many important applications in electronics. The capacitors are used for

- i. Storing electric potential energy such as batteries.
- ii. Filtering out unwanted frequency signals
- iii. Delaying voltage changes when coupled with resistors.
- iv. Used as a sensing device.
- v. Used in the audio system of the vehicle.
- vi. Used to separate AC and DC.

They have moving and fixed plates to determine the capacitance and are generally used in circuit of Transmitters and Receivers, Transistor Radios etc.

The main function of a capacitor is to store electric energy in an electric field and release this energy to the circuit as and when required. Capacitors are one of three fundamental electronic components that form the foundation of a circuit – along with resistors and inductors.

A capacitor in an electrical circuit behaves as a charge storage device. It holds the electric charge when we apply a voltage across it, and it gives up the stored charge to the circuit as when required.

4.5.5 DIODE

Diodes are used to protect circuits by limiting the voltage and to also transform AC into DC. Semiconductors like silicon and germanium are used to make most of the diodes. Even though they transmit current in a single direction, the way with which they transmit differs. There are different kinds of diodes and each type has its own applications.

Diodes can be made of either of the two semiconductor materials, silicon and germanium. When the anode voltage is more positive than the cathode voltage, the diode is said to be forward-biased, and it conducts readily with a relatively low-voltage drop. Likewise, when the cathode voltage is more positive than the anode, the diode is said to be reverse-biased. The arrow in the diode symbol represents the direction of conventional current flow when the diode conducts.

When an electric current between the electrodes passes through this diode, light is produced. In other words, light is generated when a sufficient amount of forwarding current passes through it. In many diodes, this light generated is not visible as there are frequency levels that do not allow visibility. LEDs are available in different colours. There are tricolour LEDs that can emit three colours at a time. Light colour depends on the energy gap of the semiconductor used.

Instead, a diode will have negligible resistance in one direction (to allow current flow), and very high resistance in the reverse direction (to prevent current flow).

A diode is effectively like a valve for an electrical circuit. Semiconductors are the most common type of diode.

4.6 SOFTWARE REQUIREMENTS

4.6.1 PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code,[25][26] notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.[27] Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.[28] Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software[29] and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python was conceived in the late 1980s,[30] and its implementation began in December 1989[31] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL)[32] capable of exception handling and interfacing with the Amoeba operating system.[6] Van Rossum remains Python's principal author. His continuing central role in Python's development is reflected in the title given to him by the Python

community: Benevolent Dictator For Life (BDFL). On the origins of Python, Van Rossum wrote in 1996:[33]

In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).— Guido van Rossum

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.

It is a major revision of the language that is not completely backward-compatible with previous versions.[35] However, many of its major features have been backported to the Python 2.6.x[36] and 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.[37] Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.[38][39].

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and Python 3.7.0b1 (PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale). In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.

4.6.2 ARDUINO IDE

The development of electronics is now easier thanks to arduino software (IDE), and arduino boards (hardware) . This set help to build digital and interactive devices with the help of other components. In Previous article we talk about arduino boards. In this article we will recognize what is arduino software (IDE), and how use it.

The arduino software (IDE) is an open source software, which is used to programme the Arduino boards, and is an integrated development environment, developped by arduino.cc. Allow to write and upload code to arduino boards. And it consiste of many libraries and a set of examples of mini projects.

Arduino software (IDE) is compatible with different operating systems (Windows, Linux, Mac OS X), and supports the programming languages (C/C++).

The Arduino software is easy to use for beginners, or advanced users. It uses to get started with electronics programming and robotics, and build interactive prototypes.

So Arduino software is a tool to develop new things, and create new electronic projects, by Anyone (children, hobbyists, engineers, programmers, ... etc).

4.6.3 EMBEDDED C

Embedded C is a programming language that is used in the development of Embedded Systems. Embedded Systems are specialized systems designed to perform very specific functions or tasks. Embedded System is the combination of hardware and software and the software is generally known as firmware which is embedded into the system hardware. Embedded C is used to program a wide range of microcontrollers and microprocessors. Embedded C requires less number of resources to execute in comparison with high-level languages such as assembly programming language.

Embedded C has some additional data types and keywords. There are some special datatypes in Embedded C like sbit, sfr which are used for addressing special function registers in memory. Embedded C allows us to work with hardware devices like sensors, and input-output devices. There are various Embedded C compilers to compile the embedded C program such as Keil Compiler, SPJ Compiler, Embedded GNU C Compiler, etc. Embedded Systems can be classified into small-scale, medium-scale, and sophisticated embedded systems. The devices like air conditioners, printers, and mobile phones that we use in our daily lives are programmed by embedded C.

In Embedded C we can create an efficient code to optimize the limited resources available in embedded systems. It aims to minimize memory usage and maximize performance.

Direct Hardware Interaction: Embedded C allows programmers to interact directly with hardware components, such as microcontrollers, sensors, actuators, and other peripherals. This direct interaction facilitates precise control over the hardware, critical in embedded applications.

Low-level Programming: Embedded C involves low-level programming, which deals with hardware-specific details like memory addresses, I/O ports, and register manipulation. This level of control is essential for efficiently managing hardware resources.

Real-time Operations: Embedded systems often operate in real-time environments, requiring precise timing and response to events. Embedded C allows programmers to handle real-time tasks efficiently.

CHAPTER-5

RESULTS AND PERFORMANCE ANALYSIS

5.1 PROGRAM CODE

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "ssid";
const char* WIFI_PASS = "password";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("CAPTURE FAIL");
        server.send(503, "", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
        static_cast<int>(frame->size()));

    server.setContentLength(frame->size());
    server.send(200, "image/jpeg");
    WiFiClient client = server.client();
    frame->writeTo(client);
}

void handleJpgLo()
{
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg();
}

void handleJpgHi()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
}
```

```

    }
    serveJpg();
}

void handleJpgMid()
{
    if (!esp32cam::Camera.changeResolution(midRes)) {
        Serial.println("SET-MID-RES FAIL");
    }
    serveJpg();
}

void setup(){
    Serial.begin(115200);
    Serial.println();
    {
        using namespace esp32cam;
        Config cfg;
        cfg.setPins(pins::AiThinker);
        cfg.setResolution(hiRes);
        cfg.setBufferCount(2);
        cfg.setJpeg(80);

        bool ok = Camera.begin(cfg);
        Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
    }
    WiFi.persistent(false);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
    Serial.print("http://");
    Serial.println(WiFi.localIP());
    Serial.println(" /cam-lo.jpg");
    Serial.println(" /cam-hi.jpg");
    Serial.println(" /cam-mid.jpg");

    server.on("/cam-lo.jpg", handleJpgLo);
    server.on("/cam-hi.jpg", handleJpgHi);
    server.on("/cam-mid.jpg", handleJpgMid);

    server.begin();
}

void loop()
{
    server.handleClient();
}

```

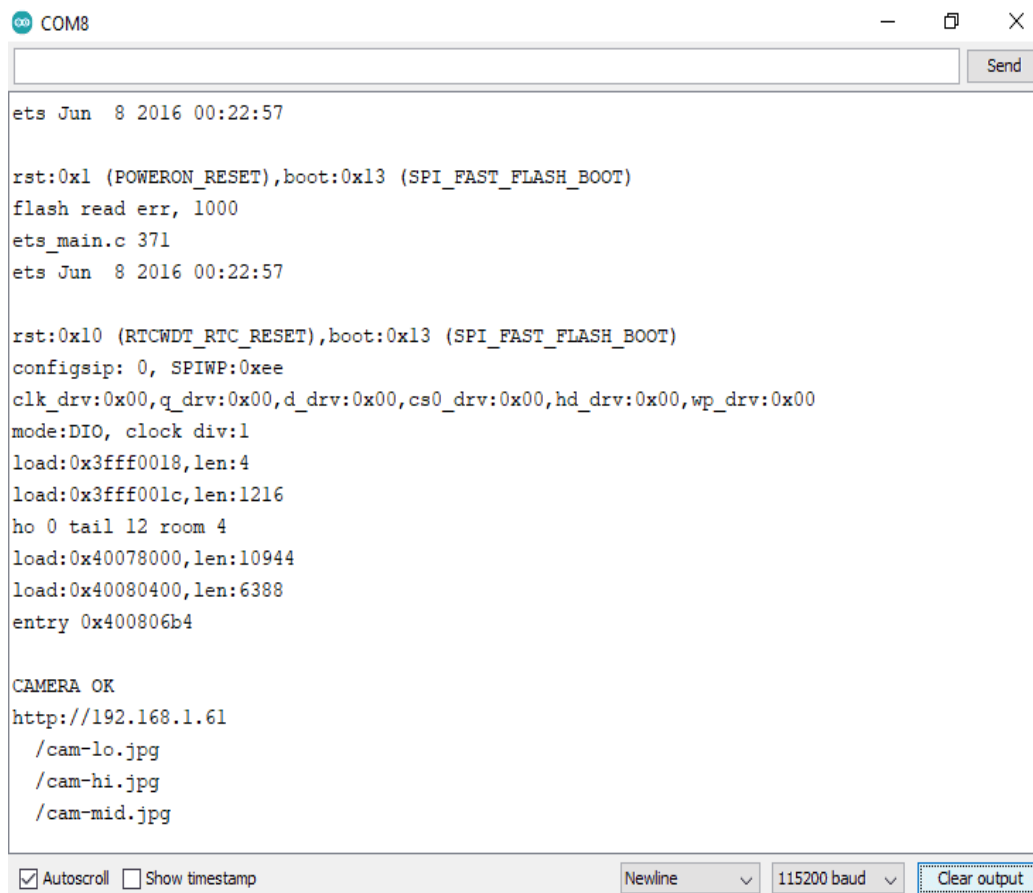
5.2 WORKING OF PROJECT

Before Uploading the code you have to make a small change to the code. Change the SSID and password variable and in accordance with your WiFi network.

Now compile and upload it to the ESP32 CAM Board. But during uploading, you have to follow few steps every time.

- Make sure the IO0 pin is shorted with the ground when you have pressed the upload button.
- If you see the dots and dashes while uploading press the reset button immediately
- Once the code is uploaded, remove the IO1 pin shorting with Ground and press the reset button once again.
- If the output is the Serial monitor is still not there then press the reset button again.

Now you can see a similar output as in the image below.



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4

CAMERA OK
http://192.168.1.61
/cam-lo.jpg
/cam-hi.jpg
/cam-mid.jpg
```

Fig 5.1 Code

Here, copy the IP address visible, we will be using it to edit the URL in python code

Now in order to proceed further, we need to install Visual studio. We are doing so because there are certain dependencies that we will be required, to install the libraries, later in this tutorial. To install go to this Visual Studio website. Then, download the latest community version.

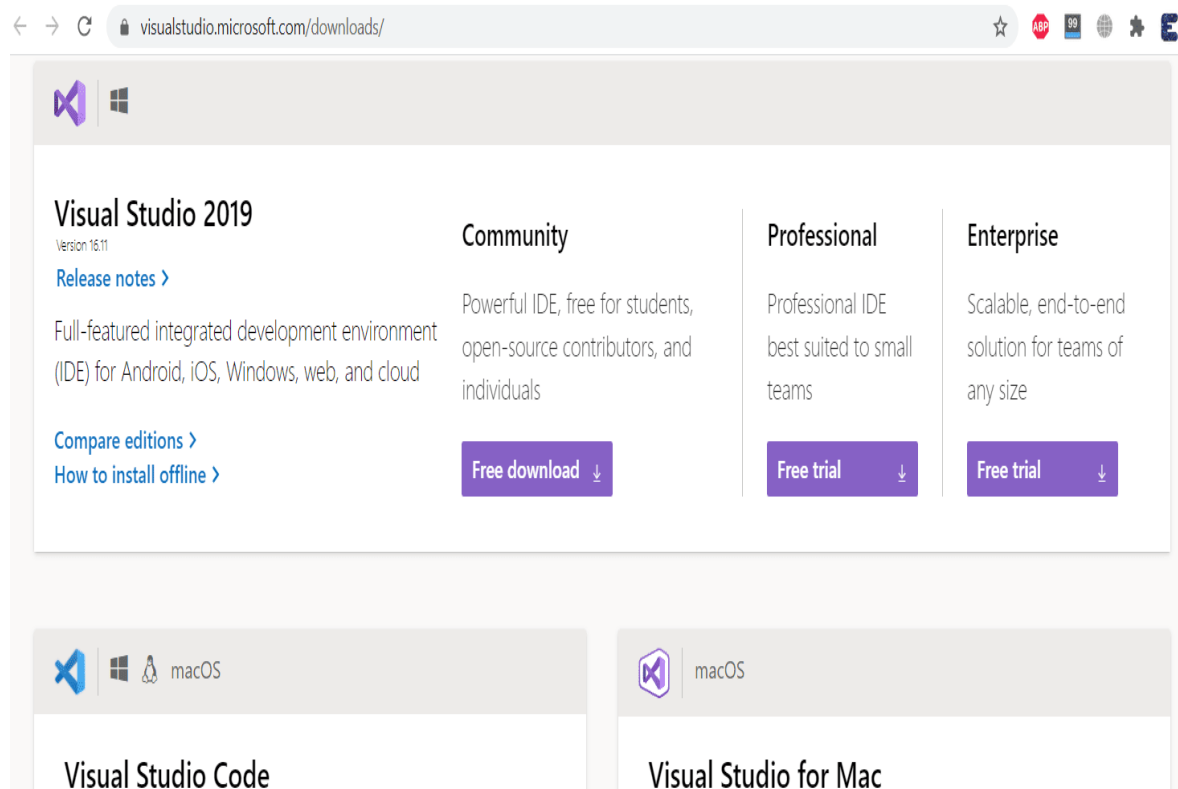


Fig 5.2 Visual Studio

Once downloaded install the software a welcome installation screen would appear, now select the community version.

After this from numerous selecting boxes, select Desktop development with C++. At the right-hand side select the boxes as in the image below.

Now that you have selected the click on Install/Modify at the Right corner below. This process will take time as it will download large files.

Once installed it will ask for restarting the PC. So that will be done automatically.

Now open the downloaded zip folder from the link below & Extract it.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

D:\python\attendace>pip install -r requirements.txt
```

Fig 5.3 Zip Folder

And press **enter**. All the necessary files will be installed.

Now we need to add the **users** who need to be detected, in the `image_folder` folder(*this is inside the zip folder that you downloaded*).

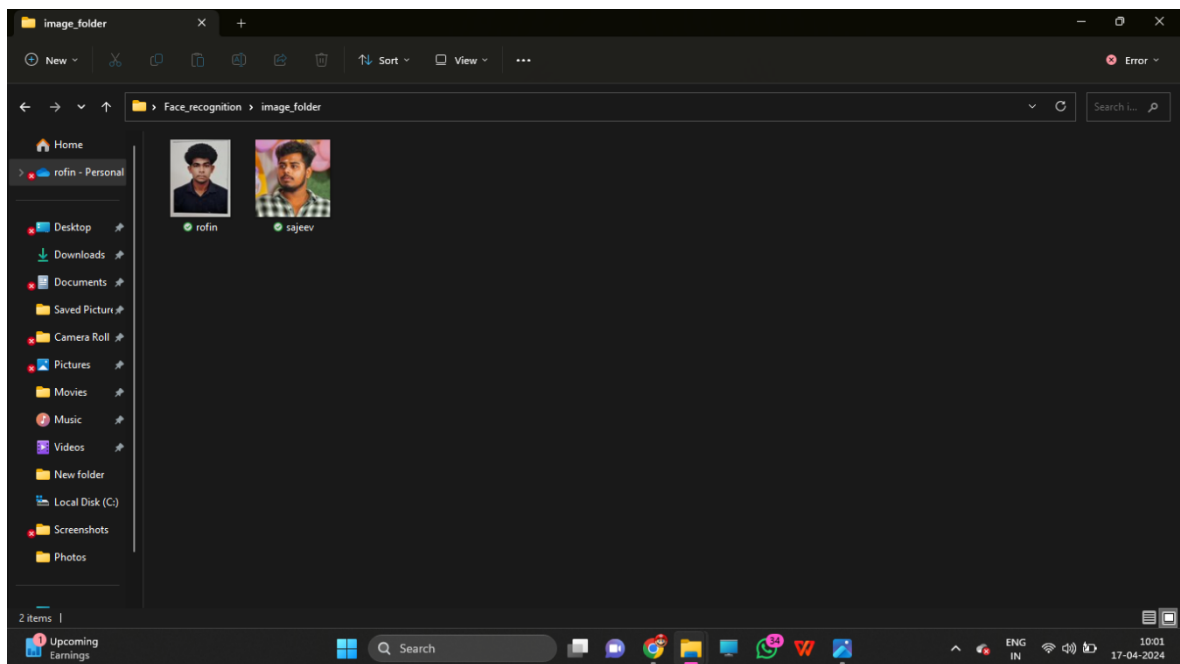


Fig 5.4 Image Folder

First, we need to update the **URL variable** in the code with the URL copied from Arduino serial monitor earlier.

Secondly, update the **path variable** in the code with the path of the **image_folder** folder.

Now we are good to go. So run the code & Stand in front of the **ESP32 Camera** by showing the face directly to Camera.

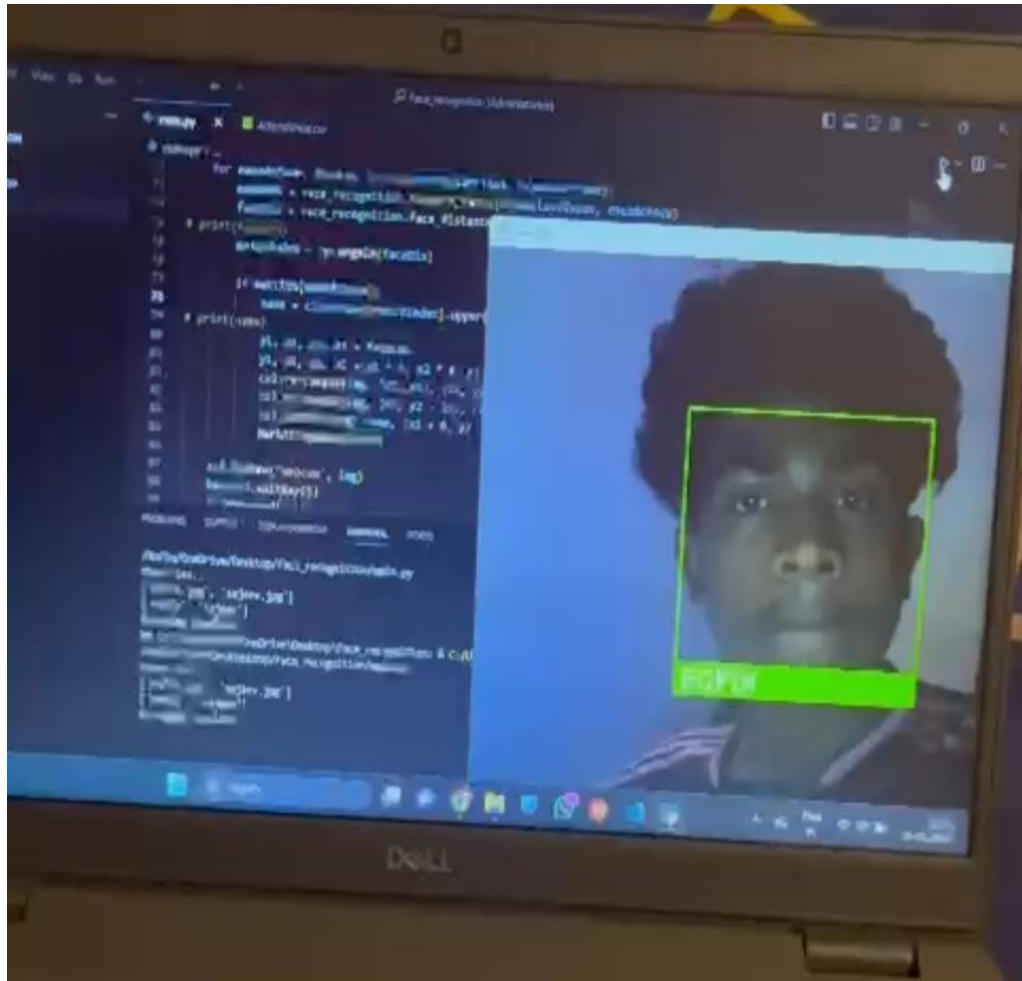


Fig 5.5 ESP 32 CAM

Congrats the face has been detected.

Now press '**q**' to close.

Now open the **Attendace.csv** file, it is in the current working directory. Here you will get all the information of people who were detected and at what time.

5.3 OUTPUT

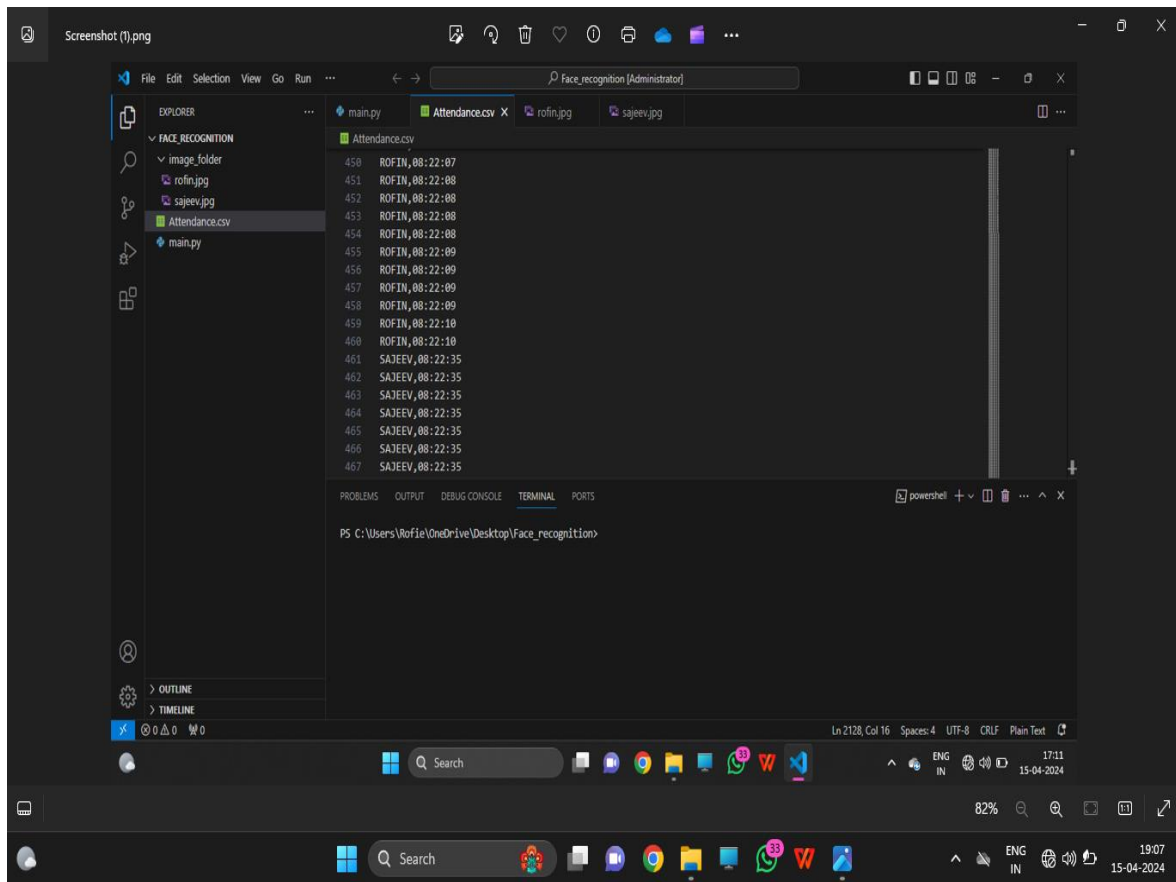


Fig 5.6 Output

Now, this **ESP32 CAM Face Recognition Based Attendance System** can be used as an actual project where you are not just detecting the students or attendees but also storing their **attendance** in an **excel** file.

CHAPTER-6

CONCLUSION

Different classrooms and offices may have different lighting setups, seating configurations, and atmosphere. The system will be tested under the majority of these circumstances, and in the majority of occasions, it has demonstrated maximum accuracy. Students may also depict various facial expressions, hairstyles, beards, spectacles, and other accessories.

To achieve a high level of accuracy and effectiveness, all of these instances are taken into account and need to be tested. This discussion leads to the conclusion that a manual and unreliable system has been replaced by one that is trustworthy, secure, quick, and efficient. A manual and unreliable system has been replaced with one that is reliable, secure, quick, and efficient.

The management of attendance and leaves will improve with the use of this system. The method will save time, lessen the amount of work that needs to be done by the administration, replace stationary with electronic equipment, and cut down on the number of people needed for the job.

6.1Result

There are two types of speeds that are produced by differing camera resolutions: speeds measured in milliseconds (ms) and frames per second (fps). You can view the detailed findings of the comparison of the several camera types utilized in the ESP-Cam [HERE](#). This is the posted photo displays the camera's OV2640 speed in real time; the resolution size, speed of motion (ms), and frames per second are all comparable.

When the camera recognizes a face, real-time data is captured and stored in Table 5. Although the speed of motion (ms) varies, it is evident that the two OV2640 camera resolutions—UXVGA and SXVGA—have the highest average speeds of motion (ms), at 83 and 87 ms, respectively. Other forms of camera resolution are essentially the same.

Numerous earlier investigations have demonstrated the ESP-32 Cam's inadequate performance. based on the low frame rates (fps), which range from 0.2 to 0.5 fps. Wireshark analysis is used to observe the data bits transmitted to the

server by employing CIF resolution throughout the video streaming process. The data throughput is shown in Figures 13 and 14, and the analysis procedure took several seconds.

6.2 Summary

Face recognition-based attendance using ESP32-CAM utilizes the ESP32-CAM module to capture images of individuals' faces, which are then processed using a face recognition algorithm. This system automates attendance tracking by detecting faces, recognizing them against a database, and marking attendance accordingly.

While offering advantages such as automation and potential scalability, challenges include ensuring accuracy in various conditions and addressing privacy concerns. Overall, it represents a promising application of embedded systems and computer vision for streamlining attendance management.

6.3 Future work

To identify the video frame's facial component. To take the identified face and extract its useful aspects. To categorize the characteristics in order to identify the observed face. To document the designated student's attendance. Any firm may manage their attendance system more successfully with the help of this project. The goal of the face recognition attendance management system is to solve issues with the existing manual methods.

Time will be saved, less work for the administration to do, stationery will be replaced with electronic equipment, and less people will be required overall thanks to the technology. This architecture makes it easy to address the problem of proxies and students being marked present when they are not. Because there is no touching involved in this procedure, also lessens the chance of pathogens spreading.

REFERENCES

- [1] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.2
- [2] P. Breuer, K.-I. Kim, W. Kienzle, B. Scholkopf, and V. Blanz. Automatic 3d face reconstruction from single images or video. In *Proc. of the 8th IEEE International Conference on Automatic Face Gesture Recognition (FG '08)*, pages 1–8, 2008.2
- [3] Chaoyang, L. Wang, Y. Wang, F. Matsushita, K., and Soong. Binocular photometric stereo acquisition and reconstruction for 3d talking head applications. In *Interspeech 2013 sub-mission*, 2013.3
- [4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 2d modeling of indoor environments. *Proc. of International Symposium on Experimental Robotics*, 2010.1
- [5] M. Hernandez, J. Choi, and G. Medioni. Laser scan quality 3-d face modeling using a low-cost depth camera. *Proc. of the 20th European Signal Processing Conference (EU-SIPCO)*, pages 1995–1999, 2012.1,2,3,4,5,7
- [6] C. Mayer, M. Wimmer, M. Eggers, and B. Radig, “Facial Expression Recognition with 3D Deformable Models.” *Second International Conferences on Advances in Computer-Human Interactions*, pp 26-31, 2009.
- [7] B. Moghaddam, J. Lee, F. Fister, and R. Machiraju, “Model-Based 3D Face Capture with Shape-from-Silhouettes.” *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.
- [8] V. Planz, and T. Vetter, “Face Recognition Based on Fitting a 3D Morphable Model”, *IEEE Trans. PAMI*, vol. 25, no. 9, pp. 1063-1074, 2003.
- [9] X. Lu, and A. Jain, “Deformation Modeling for Robust 3D Face Matching.” *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 8, pp. 1347-1356, 2008.
- [10] Z. Riaz, C. Mayer, M. Wimmer, M. Beetz, and B. Radig, “A Model Based Approach for Expression Invariant Face Recognition.” *ICB*, pp 289-298, 2009.
- [11] B. Weyrauch, and J. Huang, “Component-based Face Recognition with 3D Morphable Models.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2004.

- [12] J. Harguess, S. Gupta, and J. Aggarwal, "3D Face Recognition with the Average-Half-Face." IEEE Explore, 2008.
- [13] S. Ramanathan, A. Kassim, Y. Venkatesh, and W. Wah, "Human Facial Expression Recognition Using 3D Morphable Model." ICIP, pp 661-664, 2006.
- [14] J. Lee, B. Moghaddam, F. Fister, and R. Machiraju, "Finding Optimal Views for 3D Face Shape Modeling." Mitsubishi Electric Research Laboratories, Inc, Massachusetts, USA, 2004.
- [15] Agisoft. Agisoft photoscan. <http://www.agisoft.com/>, [Online; accessed 12-April-2017]. Balletti, C., Guerra, F., Tsioukas, V., Vernier, P., 2014. Calibration of Action Cameras for Photogrammetric Purposes. Sensors, 14(9), pp. 17471-17490. Brown, D., 1966.
- [16] Decentering distortion of lenses. Photometric Engineering 32(3), pp. 444-462. Cambi, N., 1993. Bilješke uz kipove Kibele (Magna Mater) iz Senja. Senjski zbornik, 20(1), pp. 33-43. Carrillo, L., López, A., Lozano, R., Pégard, C., 2012.
- [17] Combining stereo vision and inertial navigation system for a quad-rotor UAV. Journal of Intelligent & Robotic Systems, 65(1-4), pp. 373-387. CloudCompare.
- [18] Cloud Compare. <http://www.danielgm.net/cc/>, [Online; accessed 12-April-2017]. Ducrot, A., Dumortier, Y., Herlin, I., Ducrot, V., 2011.
- [19] Realtime quasi dense two-frames depth map for autonomous guided vehicles. In Intelligent Vehicles Symposium (IV), IEEE, pp. 497-503. Gašparović, M., Gajski, D., 2016a.
- [20] Two-step camera calibration method developed for micro UAV's. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic, Vol. XLI, Part B1, pp. 829-833. Gašparović, M., Gajski, D., 2016b.