

Face Recognition Bot Using Iot

Dr.R.Senthil Kumar, S.Gunasekar, M.Vishwanathan,
N.Vetrivel, V.Soundarya,

¹Professor & Head, Department of Mechatronics Engineering, Muthayammal Engineering College, Rasipuram,
Tamil Nadu, India -637408

²Assistant professor, Department of Mechatronics Engineering, Muthayammal Engineering College,
Rasipuram,
Tamil Nadu, India -637408

^{3,4,5}UG Student, Department of Robotics And Automation Engineering, Muthayammal Engineering College,
Rasipuram, Tamil Nadu, India -637408

Date of Submission: 25-03-2024

Date of Acceptance: 05-04-2024

Ensuring student attendance is a crucial responsibility in the classroom. When done by hand, the class usually loses a lot of useful time. This suggested remedy for the automation of the attendance system by facial recognition is the current issue. Every human being may be identified primarily by their face. This tutorial explains how to use a Raspberry Pi to detect and recognise faces in real time. This project demonstrates an effective approach that makes use of the OpenCV open source image processing library. The five components of our method are the Attendance Database, Face capturing, Face Training, Face Recognition, and Face Detection. The purpose of the face database is to identify student faces. The student database, which consists of all the student faces, is used to train the algorithm at first. Numerous more applications where facial recognition is used for authentication might benefit from this study. Using a Raspberry Pi to collect attendance reduces both the product's cost and its usefulness because it can be linked to any device. This project employs a modified version of Viola-Jones' Haar's Cascades method for face detection, LBP histograms for facial identification, and Adafruit cloud service for Internet of Things applications to refresh the database.

I. INTRODUCTION

Attendance tracking in educational institutions can be a tedious and time-consuming task. Traditional manual methods require much effort and are prone to errors. To address this problem, a new solution has emerged that automatic attendance systems using face recognition technology. This innovative approach uses the human face as a unique identifier to create a streamlined attendance management system. In

this project, a real-time face recognition system is implemented using Raspberry Pi, a versatile single-board computer, and an advanced algorithm based on the Open CV framework. Open CV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a wide range of tools and functions for image processing, computer vision, and machine learning applications. Open CV is written in C++ and has interfaces for various programming languages, including Python. Face recognition is a popular application of computer vision, and Open CV offers a comprehensive set of tools for building face recognition systems.

The system consists of five modules: Face Detection, Face Preprocessing, Face Training, Face Recognition, and Attendance Database. The process begins with the collection of facial data to create a comprehensive student database. This database serves as the training ground for the system, enabling it to recognize the faces of enrolled students seamlessly. Raspberry Pi plays a central role in this project, contributing to its cost-effectiveness and versatility. The system can connect seamlessly with various devices, making it suitable for different environments. A modified algorithm based on the Cascades method proposed by Viola-Jones for face detection, coupled with the use of Local Binary Pattern (LBP) histograms for face recognition, ensures accurate identification. MySQL is a popular open-source relational database management system (RDBMS) that is commonly used to store and manage structured data. When it comes to face recognition systems, MySQL can play a crucial role in handling and organizing the data associated with FaceRecognition.



Face Recognition Based Authentication System

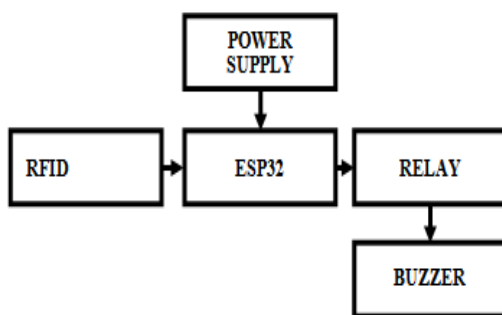
To manage the database, the system uses SQLite, a lightweight version of SQL specifically tailored for the Raspberry Pi, and MySQL for database updates. The integration of the Internet of Things (IoT) with face recognition technology has revolutionized traditional attendance systems. This introduction will explore the combination of these technologies using the versatile Raspberry Pi platform to create an efficient and secure attendance management system. The result is a Face Recognition Bot using IoT (Internet of Things) technology. Sudheer B, et.al.(1) Attendance is one of the important parameters to check the regularity and efficiency of an employee or a student. This shows the student's commitment and dedication to his work. Almost all the companies and educational institutions pose strict rules based on attendance. The conventional method that is used in colleges and schools or any institutes to mark attendance is by individually calling each student's name and marking his/her presence which is a very used way. Smitha, et.al.(2) Traditional method of attendance marking is a tedious task in many schools and colleges. It is also an extra burden to the faculties who should mark attendance by manually calling the names of students which might take about 5 minutes of entire session. This is time consuming. There are some chances of proxy attendance. Prof. Mohini Mohan Sawarkar1, et.al.(3) Taking attendance manually for a class of almost 60-80 students can be a time-consuming task if thought of it in a long run. To save the hassle and time, biometric systems are used extensively nowadays. Each person has a unique biometric feature such as fingerprint, face structure, iris, voice, etc. Mekala V, et.al.(4) Maintaining the attendance is very important in

all the institutes. Every educational institute and office have their own method of taking attendance either manually similar to old paper or file-based approach and few office and institute have adopted new methods of automatic attendance with the biometric techniques. Tippavajhala Sundar Srinivas, et.al.(5) Education institutes these days are concerned about the consistency of students' performance. One explanation for this decrease in student performance is the inadequate attendance. Dr. Manjunath B E, et.al.(6) Education institutes these days are concerned about the consistency of students' performance. One explanation for this decrease in student performance is the inadequate attendance. The longestablished attendance was taken manually that is incredibly time overwhelming and infrequently results in human error. D. Narendar Singh, M, et.al.(7) This is developed to mark the attendance for students and faculty without any person interference that makes very useful for institutions and schools to mark the attendance easily. Kumar, et.al.(8) (Senthilkumar et al. 2016) has been numerically studied about, the spherical shape dimpled tube through the Triple concentric tube heat exchanger and compared with plain tube via CFD Simulation. Perumal, S, et.al.(9) Studied about different heat transfer enhancement techniques by using different types of nanofluids such as AL_2O_3 , CUO_2 ...etc and studied about various thermal behaviours like heat transfer rate, pressure drop, friction factor, pressure drop, effectiveness and heat transfer coefficient. K.Senthilkumar, et.al.(10) has been numerically studied on dimpled concentric double tube heat exchanger with AL_2O_3 nanofluid and compared with smooth tube. Perumal saravanan, et.al.(11) numerically and experimentally studied about, the waste heat recovery from diesel engine exhaust gas through triple concentric tube heat exchanger with plain tube for various engine load conditions. Ravi Shankar Saxena, et.al.(12) The idea for this project first occurred to us in class when we observed the amount of time that must be skipped for attendance and the students' laziness after marking their attendance, which causes the method to be further delayed. Prajakta Bagul, et.al.(13) Biometrics is an evolving technology used for identification. Biometrics is associated with automatic identification of a human being depending upon biological factors as in fingerprint, iris, facial recognition. Sandesh Kulkarni, et.al.(14) A home security is the prime concern to anybody. A home security system is required to keep our belongings and valuables safe from any kind of robbery or theft. Ravi Kishore Kodali, et.al.(15) Wireless Home security and Home automation are

the dual aspects of this project. The currently built prototype of the system sends alerts to the owner over voice calls using the Internet if any sort of human movement is sensed near the entrance of his house and raises an alarm optionally upon the user's discretion. Shaik Anwar,et.al.(16) Many countries are gradually adopting smart home security control system. Today most of the home and office appliances that we interact with contain microprocessors. All of these appliances have some user interface, but many users become frustrated with the difficulty of using the complex functions of their appliances.

II. EXISTING SYSTEM

Facial recognition technology is a biometric method that identifies and verifies individuals based on their unique facial features. This technology is utilized in various applications such as security access systems, attendance tracking, and more. It utilizes algorithms to analyze and compare facial patterns, making it a reliable method for authentication. Our Face Recognition Bot is further enhanced by the integration of Radio-Frequency Identification (RFID) technology, which provides a convenient and secure means of identification. The ESP32 CAM module, a powerful microcontroller with integrated Wi-Fi and a camera, is used to implement the facial recognition feature. The Excel spreadsheet acts as a centralized database, storing information about recognized individuals, their access history, and RFID tag associations. This integration allows for easy tracking and analysis of data.



Existing Method Block Diagram

BLOCK DIAGRAM EXPLANATION

Each individual is assigned a unique RFID tag, and their information (name, RFID tag ID) is stored in the Excel database. Capture facial images of individuals using the ESP32 CAM and store them locally or remotely. Associate each face image with the corresponding RFID tag ID in the database. Excel is used as a database to store

information about individuals, including their names and corresponding RFID tag IDs. Implement logic to make decisions based on the face recognition results. If a face is recognized, retrieve the associated RFID tag ID, look up the corresponding information in the Excel database, and make decisions accordingly. A relay is used to control an external device, such as an electronic door lock, based on the face recognition results.

ESP32 MICROCONTROLLER

ESP32 is the name of the chip that was developed by Expressive Systems. This provides Wi-Fi (and in some models) dual-mode Bluetooth connectivity to embedded devices. While ESP32 is technically just the chip, modules and development boards that contain this chip are often also referred to as "ESP32" by the manufacturer.



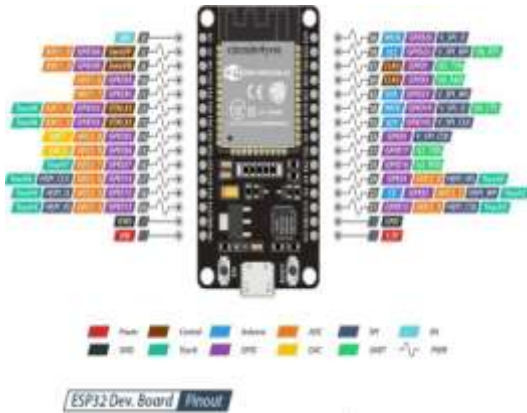
ESP 32 Wi-Fi module

The ESP32 chip has a Ten silica Xtensa LX6 microprocessor in both dual-core and single-core variations, with a clock rate of over 240 MHz.

Features and specifications

S.NO	ESP32	DESCRIPTION
1	Core	2
2	Architecture	32 bits
3	Clock	Ten silica xtensa LX-106 160_240MHZ
4	Wi-Fi	IEEE802.11 b/g/n
5	Bluetooth	Yes - classic & BLE
6	RAM	520KB
7	Flash	Extern QSPI - 16MB
8	GPIO	2
9	DAC	22
10	ADC	18
11	Interfaces	SPI-I2C-UART- I2S-CAN

ESP 32 PINOUT



ESP32 pinout configuration

ESP32 Peripheral features

1. 18 Analog-to-Digital Converter (ADC) channels
2. 10 Capacitive sensing GPIOs
3. 3 UART interfaces
4. 3 SPI interfaces
5. 2 I2C interfaces
6. 16 PWM output channels
7. 2 Digital-to-Analog Converters (DAC)
8. 2 I2S interfaces

GPIO pins

ESP32 Wroom32 Devkit has total 25 GPIOs out of that few pins are Input only Pins,

Input Only Pins

GPIO 34, GPIO 35, GPIO 36, GPIO 39

Not all pins have input pullup, you need external pullup on these pins when using as input pullup.

Pins with internal pull up INPUT_PULLUP

GPIO14, GPIO16, GPIO17, GPIO18, GPIO19, GPIO21, GPIO22, GPIO23

Pins without internal pull up

GPIO13, GPIO25, GPIO26, GPIO27, GPIO32, GPIO33

Analog Input Pins

Analog to digital conversion is the ability to read a voltage level found on a pin between 0 and some maximum value and convert that analog value into a digital representation. Varying the voltage applied to the pin will change the value read. The ESP32 has an analog to digital converter built into it with a resolution of up to 12 bits which is 4096 distinct values.

- ADC1_CH0
- ADC1_CH3
- ADC1_CH4

- ADC1_CH5
- ADC1_CH6
- ADC1_CH7

Capacitive touch pins

The ESP32 has 10 internal capacitive touch sensors.

- T0 (GPIO 4)
- T1 (GPIO 0)
- T2 (GPIO 2)
- T3 (GPIO 15)
- T4 (GPIO 13)
- T5 (GPIO 12)
- T6 (GPIO 14)
- T7 (GPIO 27)
- T8 (GPIO 33)
- T9 (GPIO 32)

Digital to Analog Converter (DAC)

There are 2 x 8 bits DAC channels on the ESP32 to convert digital signals into analog voltage signal outputs.

- DAC1 (GPIO25)
- DAC2 (GPIO26)

RTC GPIOs

There is RTC GPIO support on the ESP32. The GPIOs routed to the RTC low-power subsystem can be used when the ESP32 is in deep sleep. These RTC GPIOs can be used to wake up the ESP32 from deep sleep when the Ultra-Low Power (ULP) co-processor is running. The following GPIOs can be used as an external wake up source.

- RTC_GPIO0 (GPIO36)
- RTC_GPIO3 (GPIO39)
- RTC_GPIO4 (GPIO34)
- RTC_GPIO5 (GPIO35)
- RTC_GPIO6 (GPIO25)
- RTC_GPIO7 (GPIO26)
- RTC_GPIO8 (GPIO33)
- RTC_GPIO9 (GPIO32)
- RTC_GPIO10 (GPIO4)
- RTC_GPIO11 (GPIO0)
- RTC_GPIO12 (GPIO2)
- RTC_GPIO13 (GPIO15)
- RTC_GPIO14 (GPIO13)
- RTC_GPIO15 (GPIO12)
- RTC_GPIO16 (GPIO14)
- RTC_GPIO17 (GPIO27)

PWM

The ESP32 LED PWM controller has 16

independent channels that can be configured to generate PWM signals with different properties. All pins that can act as outputs can be used as PWM pins (Input only pin GPIOs 34 to 39 can't generate PWM).

Serial

ESP32 has three serial ports
 First Serial RX0, TX0 is used for programming,

- GPIO3 (U0RXD)
- GPIO1(U0TXD)

Another Serial port is available on

- GPIO16 (U2RXD).
- GPIO17 (U2TXD).

When programming it is named as Serial2.

I2C

When using the ESP32 with the Arduino IDE, you should use the ESP32 I2C default pins (supported by the Wire library):

- GPIO 21 (SDA)
- GPIO 22 (SCL)

SPI

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO2 3	GPIO1 9	GPIO18	GPIO15
HSPI	GPIO1 3	GPIO1 2	GPIO14	GPIO5

Interrupts

All GPIOs can be configured as interrupts.

Enable (EN)

Enable (EN) is the 3.3V regulator's enable pin. It's pulled up, so connect to ground to disable the 3.3V regulator. This means that you can use this pin connected to a pushbutton to restart your ESP32.

ESP 32 functions

ESP32 has many applications when it comes to the Internet of Things (IoT). Here are just some of the IoT functions the chip is used for:

Networking: The module's Wi-Fi Antenna and dual-core enables embedded devices to connect to routers and transmit data.

Data Processing: Includes processing basic inputs from Analog and Digital sensors to far more complex calculations with an RTOS or Non-OS SDK.

P2P Connectivity: Creates direct communication between different ESPs and other devices using IoT P2P connectivity.

Web Server: Access pages written in HTML or development languages.

ESP32 applications

- Programmable Logic Controllers (PLCs)
- including wearable health monitors
- including HVAC and thermostats
- including surveillance cameras and smart locks applications
- Prototyping of IoT devices
- Low power battery operated applications
- Network projects
- Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

RFID TAG AND READER

The wireless, non-contact conveyance of data via radiofrequency waves is known as radio frequency identification, or RFID. Readers, also known as interrogators, are gadgets that connect with RFID tags by sending and receiving radio waves. Fixed RFID readers and mobile RFID readers are the two main categories into which RFID readers are usually classified. .. Certain readers may link to up to 32 RFID antennae by adding a multiplexer.

RFID gives an object a unique identity by using radio waves to read and collect data from a tag affixed to the object. Because RFID technology may assist organizations in many ways, it is employed in many different sectors and applications.

Features:

- Highly integrated analog circuitry to demodulate and decode responses.
- Supports ISO/IEC 14443 A/MIFARE.
- Typical operating distance in reading/Write mode up to 50 mm.
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd.
- SPI up to 10 Mbit/s.
- FIFO buffer handles 64 bytes send and receive.
- Flexible interrupt modes.
- Power-down by software mode.
- Programmable Timer.
- 2.5 V to 3.3 V power supply.
- CRC coprocessor.
- Internal self-test.

DISADVANTAGES

- RFID systems can be vulnerable to hacking or cloning, posing a security risk if not adequately protected.
- Depending on the capabilities of the ESP32 CAM and storage limitations, there might be

restrictions on the number of faces that can be stored in the database.

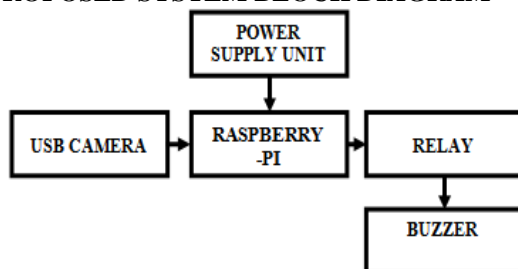
- Relying on Excel for data storage may not be the most scalable or efficient solution for larger datasets. Excel is not designed to handle large volumes of data or concurrent access from multiple sources.

III. PROPOSED SYSTEM

In an era defined by rapid technological advancements, the fusion of the Internet of Things (IoT) and artificial intelligence has paved the way for innovative solutions that redefine the boundaries of convenience and security. One such groundbreaking project emerges in the form of a "Face Recognition Bot," seamlessly integrating IoT, a USB camera, Raspberry Pi, and Telegram messaging. As the world becomes increasingly interconnected, the need for robust security measures and efficient automation has never been more crucial. The convergence of IoT and facial recognition technology offers a sophisticated solution that goes beyond traditional surveillance methods. By leveraging the power of a USB camera and the compact yet powerful Raspberry Pi microcomputer, this project aims to create a versatile and intelligent system capable of recognizing faces in real time.

The Raspberry Pi serves as the central hub, orchestrating the communication between the USB camera and the Telegram messaging platform. This compact yet highly capable device acts as the brain of the operation, executing intricate facial recognition algorithms and facilitating seamless interaction between the user and the system. Through the integration of Telegram, users can receive instant notifications and control the Face Recognition Bot remotely, enhancing the accessibility and user-friendliness of the system. In this endeavour, we delve into the intricacies of designing and implementing a Face Recognition Bot that not only enhances security but also exemplifies the potential of IoT-driven applications.

PROPOSED SYSTEM BLOCK DIAGRAM



Proposed Block Diagram

BLOCK DIAGRAM EXPLANATION

This project utilizes a USB camera to capture images or video frames as input. Typically, the camera is connected to a Raspberry Pi, a small, cost-effective, credit-card-sized computer that can be used for various purposes. The Raspberry Pi serves as the primary computing platform for this project. It is a versatile single-board computer that runs a Linux-based operating system and can be used for various projects, including image processing and computer vision tasks. The buzzer, a sound-producing component, generates audible alerts or notifications. In this context, it can signal when a face is recognized or indicate a specific event during the face recognition process. Python, a widely used programming language known for its simplicity and readability, is often employed in Raspberry Pi projects due to its ease of use and the availability of various libraries and modules for different functionalities. OpenCV, a popular open-source computer vision library, provides tools and functions for image and video processing. It includes a wide range of algorithms for tasks such as image manipulation, feature detection, and object recognition, making it suitable for face recognition applications. LBPH, a texture-based face recognition algorithm, works by extracting local binary patterns from images and creating histograms of these patterns for each face. This algorithm is often used in combination with OpenCV for facial recognition tasks.

The proposed system automates attendance for different organizations and reduces the flaws of existing manual systems. The system calculates attendance subject-wise, with the data of students and subjects manually added by the administrator. Whenever the time for a corresponding subject arrives, the system automatically starts taking snaps and checks whether human faces appear in the given image or not. Histogram of Oriented Gradient is utilized for face detection and techniques to calculate and compare 25-d face features for face recognition. Once faces are detected and recognized with the existing database, the system calculates attendance for the recognized students with the respective subject ID in real time. An Excel sheet is generated and saved by the system, which is automatically sent to the Telegram application. The system involves the Internet of Things (IoT) technology, and specifically, it utilizes Telegram messaging.

POWER SUPPLY UNIT

A 5-volt adapter is a power supply that provides a stable 5-volt DC (direct current) output. This voltage level is a standard for many electronic

devices, including the Raspberry Pi, which is a single-board computer. Raspberry Pi boards come with USB ports that can be used to connect peripherals such as keyboards, mice, and other external devices. These USB ports are designed not only for data transfer but also to supply power. For proper operation, the Raspberry Pi requires a 5-volt power supply, which is usually provided through a micro USB or USB-C connector, depending on the model. Back powering is a principle that allows a device to be powered through one of its output ports. In the context of the Raspberry Pi, this means supplying power to the board through one of its USB ports. This principle is applicable to Raspberry Pi models B and B+ and potentially earlier models, as indicated by the mention of old models up to model B+.



Power Adapter

RASPBERRY PI

The Raspberry Pi is a series of small, affordable, single-board computers that the Raspberry Pi Foundation has developed. These computers are designed to promote computer science education and facilitate the development of DIY projects and prototypes. The Raspberry Pi's central processing unit (CPU) and graphics processing unit (GPU) are integrated into a single chip, which streamlines the design, making it more compact and energy-efficient. The CPU is responsible for executing instructions and performing calculations, while the GPU handles graphics-related tasks, which is crucial for visual outputs. The compact design of the Raspberry Pi resembles a credit card-sized board, making it highly versatile and suitable for various applications, ranging from hobbyist projects to embedded systems in commercial products. The essential components like USB ports, RAM (Random Access Memory), and an SD card slot are soldered directly onto the board, which enhances the Raspberry Pi's connectivity.

Users can attach peripherals such as keyboards, mice, webcams, and external storage

devices, expanding the functionality of the mini-computer. The Raspberry Pi uses an SD card for storage, allowing for flexibility, as users can switch between different operating systems by swapping out the SD card. The Raspberry Pi is compatible with various operating systems, including Raspbian (now known as Raspberry Pi OS), allowing users to choose and install an operating system based on their specific needs. The integrated and efficient design of the Raspberry Pi contributes to its low power consumption, making it an ideal choice for projects where energy efficiency is a priority, such as in remote or off-grid applications. The Raspberry Pi has a vibrant community of developers and enthusiasts, which fosters continuous development, with a wide range of software and hardware projects available. This collaborative environment encourages learning and innovation among users.



Raspberry Pi

USB CAMERA

This camera operates through a simple yet effective mechanism. It relies on a USB cable to establish a connection with a computer. This connection serves as a bridge through which the camera's captured feeds are transmitted to the computer for further processing and utilization. The integration of hardware and software components makes this system a versatile tool for capturing, viewing, and sharing images. The fundamental process begins with physically connecting the camera to the computer using a USB cable. This cable provides a link for power and facilitates the transfer of data between the camera and the computer. Once connected, the camera becomes an extension of the computer's capabilities, leveraging the processing power and storage capacity of the host system.

The camera captures images through its lens and sensor, converting visual information into digital data. This digital data is then transferred via

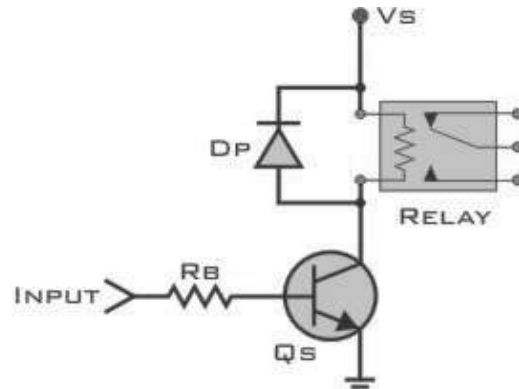
the USB cable to the computer in real-time or as captured snapshots, depending on the camera's settings. The computer recognizes the connected camera as an external device equipped with the necessary drivers. A software application plays a crucial role in interacting with the camera and its captured content. This software is designed to communicate with the camera, receive the incoming data stream, and provide a user-friendly interface for controlling various aspects of the camera's functionality. Users can access features such as adjusting settings, capturing images, and previewing the live feed directly through the software.



USB Camera

RELAY

A relay circuit is an electrically operated switch that uses an electromagnet to control the switching of a circuit mechanically. It's often used to control a high-power circuit with a low-power signal, such as when a microcontroller or a low-voltage sensor needs to control a higher-voltage device. When a voltage is applied to the base of the BC547 through the resistor, it allows current to flow from the collector to the emitter, turning the transistor on. The diode protects the transistor from back EMF generated when the relay coil is de-energized. When the transistor is on, current flows through the relay coil, activating the relay. The activated relay switches the common (COM) terminal connection from the Normally Open (NO) to the Common (COM) terminal, allowing current to flow through the load.



Relay Circuit Diagram

BUZZER

Buzzers typically consist of a coil, a magnet, and a diaphragm or membrane. When an electrical current passes through the coil, it creates a magnetic field that attracts the diaphragm towards the coil. The movement of the diaphragm produces sound waves, resulting in the characteristic buzzing or beeping sound.



Buzzer

These buzzers use the piezoelectric effect to produce sound. They contain a piezoelectric crystal that vibrates when an electric field is applied. Buzzers have specific operating voltage requirements, and the frequency of the sound they produce is determined by their design and construction.

- Integration with IoT allows for remote access and control of the face recognition system, enabling users to receive notifications or messages via Telegram when specific faces are detected.
- Python is a versatile and widely used programming language; it allows for efficient development and easy integration with various libraries and components.
- Open CV is a powerful computer vision library that provides a wide range of tools and functions for image processing and computer

vision applications, making it a suitable choice for facial recognition systems.

APPLICATION

- Implementing the system in access control systems can enhance security by allowing entry only to authorized individuals based on facial recognition.
- The system can be employed in educational institutions or workplaces for automated attendance tracking, eliminating the need for manual attendance management.
- The IoT integration enables remote monitoring and control of the system, making it suitable for surveillance applications in various environments.
- Users can receive customized notifications via Telegram, allowing for seamless integration with messaging platforms for immediate awareness.

IV. MODULES USED TO RECOGNIZE THE FACE

OPENCV

An open-source software library for computer vision and machine learning is called OpenCV, short for Open Source Computer Vision Library. It was first created by Intel and is currently maintained by the OpenCV Foundation, a community of developers. A vast open-source library for image processing, machine learning, and computer vision is called Opencv. It now has a significant impact on real-time functioning, which is critical to modern systems. It may be used to process photos and videos in order to recognise faces, objects, and even human handwriting.

Python can process the OpenCV array structure for analysis when it is combined with other libraries, such NumPy. We employ vector space and apply mathematical operations to these characteristics in order to identify an image pattern and its different features.

OpenCV allows you to perform various operations in the image.

Read the Image:OpenCV helps you to read the image from file or directly from camera to make it accessible for further processing.

Image Enhancment: You will be able to enhance image by adjusting the brightness, sharpness or contract of the image. This is helpful to visualize quality of the image.

Object detection: As you can see in the below image object can also be detected by using OpenCV , Bracelet , watch , patterns, faces can be

detected. This can also include to recognize faces, shapes or even objects.

Image Filtering: You can change image by applying various filters such as blurring or Sharpening.

Draw the Image:OpenCV allows to draw text, lines and any shapes in the images.

Saving the Changed Images: After processing, you can save images that are being modified for future analysis.

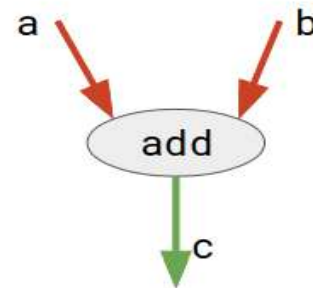
TENSORFLOW

Google created the open-source machine learning library Tensor Flow. Because Tensor flow makes it easier to create computational networks and execute them efficiently across a range of hardware platforms, it is used to develop and train deep learning models.

TensorFlow is basically a software library for numerical computation using data flow graphs where:

NODES in the graph represent mathematical operations.

EDGES in the graph represent the multidimensional data arrays (called tensors) communicated between them. (Please note that tensor is the central unit of data in TensorFlow).Consider the diagram given below:



Tensorflow node and Edge

V. INTERNET OF THINGS (IOT) INTRODUCTION

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Working of IoT

An IoT ecosystem consists of web-enabled smart devices that use embedded systems,

such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments.

IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another.

The devices do most of the work without human intervention, although people can interact with the devices for instance, to set them up, give them instructions or access the data.

Important of IoT

The internet of things helps people live and work smarter, as well as gain complete control over their lives. In addition to offering smart devices to automate homes, IoT is essential to business.

IoT provides businesses with a real-time look into how their systems really work, delivering insights into everything from the performance of machines to supply chain and logistics operations.

IoT enables companies to automate processes and reduce lab (or) costs. It also cuts down on waste and improves service delivery, making it less expensive to manufacture and deliver goods, as well as offering transparency into customer transactions.

Benefits of IoT:

- Save time and money
- Information can access from anywhere any time
- Improved communication
- Transferring the data packets
- Improve the quality of business service

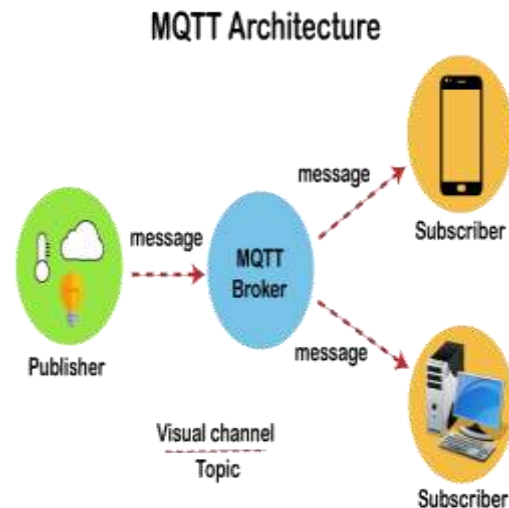
Application of IoT

- Connected vehicles
- Traffic managements
- Smart grids
- Environmental monitoring
- Smart homes/smart buildings
- Smart cities
- Industrial, agricultural and commercial management

MQTT (MQ TELEMETRY TRANSPORT)

MQTT (MQ Telemetry Transport) is a lightweight open messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information in low-bandwidth environments. The protocol, which

employs a publish/subscribe communication pattern, is used for machine-to-machine (M2M) communication.



MQTT Architecture

Created as a low-overhead protocol to accommodate bandwidth and CPU limitations, MQTT was designed to run in an embedded environment where it could provide a reliable, effective path for communication. Suitable for connecting devices with a small code footprint, MQTT is a good choice for wireless networks that experience varying levels of latency due to occasional bandwidth constraints or unreliable connections. The protocol has applications in industries ranging from automotive to energy to telecommunications.

Although MQTT started as a proprietary protocol used to communicate with supervisory control and data acquisition (SCADA) systems in the oil and gas industry, it has become popular in the smart device arena and today is the leading open-source protocol for connecting the internet of things (IoT) and industrial IoT (IIoT) devices.

While the TT in MQTT stands for Telemetry Transport, the MQ is in reference to a product called IBM MQ. Although the spell-out for MQTT is sometimes given as Message Queuing Telemetry Transport, there is no message queuing in MQTT communication.

MQTT MESSAGES

When a client wants to send data to the broker, this is known as a “publish.” When a client wants to receive data from the broker, it will “subscribe” to a topic or topics. When a client subscribes to a certain topic, it will receive all messages published on that topic going forward.

Along with the message itself, the publisher also sends a QoS (Quality of Service) level. This level defines the guarantee of delivery for the message.

These QoS levels are as follows:

At most once: When the message is published, the broker will only receive the message “at most once.” This level should not be used for mission-critical information since it carries the risk that the subscribers will not receive the message.

At least once: The publisher continues to resend the message until it receives an acknowledgment from the broker regarding the particular message. In other words, it’s more important that the message is received than it is to ensure it is only received once. This is probably the most commonly used QoS level.

Exactly once: The publisher and broker work together to ensure the broker will receive and act on a message exactly once. This requires some additional overhead in the form of a four-part handshake. Although this is the safest QoS level, it is also the slowest and therefore only used when necessary

DESCRIPTION

MQTT MESSAGES	DESCRIPTION
CONNECT	Client request to connect to server
CONNACK	Connect acknowledgement
PUBLISH	Publish message
PUBACK	Publish acknowledgement
PUBREC	Publish received
PUBREL	Publish release
PUBCOMP	Publish complete
SUBSCRIBE	Client subscribe request
SUBACK	Subscribe acknowledgement
UNSUBSCRIBE	Unsubscribe request
UNSUBACK	Unsubscribe acknowledgement
PINGREQ	Ping request
PINGRESP	Ping response
DISCONNECT	Client disconnecting

Benefits of MQTT

- Efficient data transmission and quick to implement, due to its being a lightweight protocol;

- Low network usage, due to minimized data packets;
- Efficient distribution of data;
- Successful implementation of remote sensing and control;
- Fast, efficient message delivery;
- Uses small amounts of power, which is good for the connected devices; and Optimizes network bandwidth.

Applications of MQTT

- Synchronization of sensors, such as fire detectors or motion sensors for theft detection, to determine if a hazard is valid
 - Monitoring health parameters using sensors for patients leaving a hospital
- Sensors alerting people of danger

ADAFRUIT IO

Adafruit io is a platform designed by ADAFRUIT IO to display, respond, and interact with your project’s data. Adafruit also keep your data private (data feeds are private by default) and secure (ADAFRUIT will never sell or give this data away to another company).

USAGE WITH ADAFRUIT IO

The esp32-s2 is an affordable, all-in-one, option for connecting your projects to the internet using our IOT platform, ADAFRUIT IO.

For more information and guides about ADAFRUIT IO, check out the ADAFRUIT IO basics series.

INSTALL LIBRARIES

In the Arduino ide, navigate to sketch -> include library->manage libraries. Enter Adafruit io Arduino into the search box, and click install on the Adafruit io Arduino library option to install version 4.0.0 or higher.

ADAFRUIT IO SETUP

If you do not already have an Adafruit io account, create one now. Next, navigate to the Adafruit io dashboards page.

We'll create a dashboard to visualize and interact with the data being sent between your esp32-s2 board and Adafruit io.

- Click the new dashboard button.
- Name your dashboard my esp32-s2.
- Your new dashboard should appear in the list.
- Click the link to be brought to your new dashboard.

We'll want to turn the board's led on or off from Adafruit io. To do this, we'll need to add a

- toggle button to our dashboard.
- Click the cog at the top right-hand corner of your dashboard.
- In the dashboard settings dropdown, click create new block.
- Select the toggle block.
- Under my feeds, enter led as a feed name. Click create.
- Choose the led feed to connect it to the toggle block. Click next step.

UNDER BLOCK SETTINGS,

- Change button on text to 1
- Change button off text to 0
- Click create block

VI. PROGRAMMING CODE

CAPTURE PICTURE:

```
import cv2
name = 'VISHWA-staff'
cam = cv2.VideoCapture(0)
cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
cv2.resizeWindow("press space to take a photo", 500, 300)
img_counter = 0
while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
        break
    cv2.imshow("press space to take a photo", frame)
    k = cv2.waitKey(1)
    if k%256 == 27:
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        img_name = "/home/pi/Desktop/face23324/photo/"+ name +"/image_{ }.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1
    cam.release()
    cv2.destroyAllWindows()
```

TRAIN:

```
from imutils import paths
import face_recognition

import pickle
import cv2
import os

print("[INFO] start processing faces...")
imagePaths = list(paths.list_images("photo"))

knownEncodings = []
knownNames = []

for (i, imagePath) in enumerate(imagePaths):

    print("[INFO] processing image { }/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]

    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

```
boxes = face_recognition.face_locations(rgb,
    model="hog")

encodings = face_recognition.face_encodings(rgb, boxes)

for encoding in encodings:

    knownEncodings.append(encoding)
    knownNames.append(name)

print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
f.write(pickle.dumps(data))
f.close()
```

TEST:

```
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import imutils
import pickle
import time
import cv2
import time
import board
import digitalio
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
buzzer=23
GPIO.setup(buzzer,GPIO.OUT)
from Adafruit_IO import Client
aio = Client('Facebotproject', 'aio_jCaH13si074VDVIYxm49kEoqLi6O')

#Initialize 'currentname' to trigger only when a new person is identified.
currentname = "unknown"
#Determine faces from encodings.pickle file model created from train_model.py
encodingsP = "encodings.pickle"

# load the known faces and embeddings along with OpenCV's Haar
# cascade for face detection
print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(encodingsP, "rb").read())

# initialize the video stream and allow the camera sensor to warm up
# Set the ser to the following
# src = 0 : for the build in single web cam, could be your laptop webcam
# src = 2 : I had to set it to 2 in order to use the USB webcam attached to my laptop
#vs = VideoStream(src=2,framerate=10).start()
```

```
vs = VideoStream(usePiCamera=False).start()
time.sleep(2.0)

# start the FPS counter
fps = FPS().start()

# loop over frames from the video file stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to 500px (to speedup processing)
    frame = vs.read()
    frame = imutils.resize(frame, width=500)
    # Detect the fce boxes
    boxes = face_recognition.face_locations(frame)
    # compute the facial embeddings for each face bounding box
    encodings = face_recognition.face_encodings(frame, boxes)
    names = []

    # loop over the facial embeddings
    for encoding in encodings:
        # attempt to match each face in the input image to our known
        # encodings
        matches = face_recognition.compare_faces(data["encodings"],
            encoding)
        name = "Unknown" #if face is not recognized, then print Unknown
        GPIO.output(buzzer,GPIO.HIGH)
        sleep(0.5)
        # check to see if we have found a match
        if True in matches:
            # find the indexes of all matched faces then initialize a
            # dictionary to count the total number of times each face
            # was matched
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            counts = {}

            # loop over the matched indexes and maintain a count for
            # each recognized face face
            for i in matchedIdxs:
                name = data["names"][i]
                counts[name] = counts.get(name, 0) + 1

            # determine the recognized face with the largest number
            # of votes (note: in the event of an unlikely tie Python
            # will select first entry in the dictionary)
            name = max(counts, key=counts.get)
            GPIO.output(buzzer,GPIO.LOW)
            sleep(0.5)
            #If someone in your dataset is identified, print their name on the screen
            if currentname != name:
                currentname = name
                print(currentname)

        # update the list of names
        names.append(name)
```

```

test = aio.feeds('attendance')
aio.send_data(test.key, name)
# loop over the recognized faces
for ((top, right, bottom, left), name) in zip(boxes, names):
    # draw the predicted face name on the image - color is in BGR
    cv2.rectangle(frame, (left, top), (right, bottom),
        (0, 255, 225), 2)
    y = top - 15 if top - 15 > 15 else top + 15
    cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
        .8, (0, 255, 255), 2)

# display the image to our screen
cv2.imshow("Facial Recognition is Running", frame)
key = cv2.waitKey(1) & 0xFF
# quit when 'q' key is pressed
if key == ord("q"):
    break

# update the FPS counter
fps.update()

googggoo
# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
test = aio.feeds('attendance')
aio.send_data(test.key, name)
# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()

```

VII. RESULT AND DISCUSSION.



Overall Assembling Of Face Recognition Bot Using IoT

The IoT and facial recognition based automatic attendance system hardware which consists Raspberry Pi 4 , USB camera, Buzzer. The Raspbian OS installed in SD Card (16 GB, C10). The python script written by using Thonny IDE as

a python compiler.



Expected Facial Recognition

The python script with .py extension, totally three files in our work for capture the person face image. the database created with help of this code. The trained harscascaded model created with help of trained code. The recognition and attendance placed achieved with testing.py code.

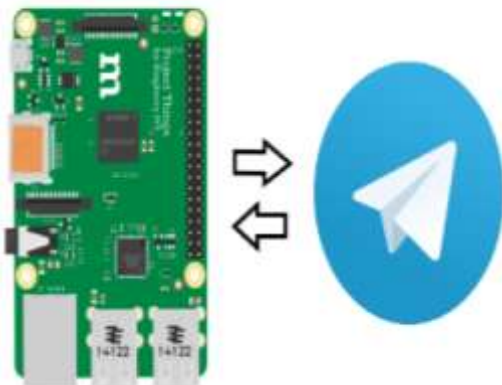


Expected Telegram Notification

IOT (Telegram)

The integration of IoT suggests that the face recognition system is connected to the internet and potentially interacts with other devices. IoT devices could include cameras, sensors, or any other smart devices that contribute to the functionality of the system. Telegram is a popular messaging platform with a Bot API that allows developers to create bots capable of interacting with users. The system may use Telegram to send notifications, alerts, or other messages related to face recognition events.

Sharing texts and files



IoT Communication between Raspberry Pi and Telegram

ADVANTAGES

- The use of the Local Binary Pattern Histogram Algorithm enhances facial recognition accuracy, making it robust against variations in lighting and facial expressions.

VIII. CONCLUSION.

Facial recognition technology has become increasingly popular in recent years, and the Local Binary Pattern Histogram Algorithm is one of the most accurate methods for identifying individuals. By integrating this algorithm with Open CV and using Python as the programming language, we can ensure seamless integration and efficient code

implementation. Our attendance system is designed to log and maintain records automatically, reducing the need for manual entry and minimizing errors associated with traditional attendance tracking methods. We have also created a cost-effective and accessible hardware setup by combining a USB camera with a Raspberry Pi, which captures facial images in real time. To enhance user interaction and provide immediate feedback, we have included a buzzer as a physical indicator. Additionally, we have leveraged IoT for communication with Telegram, which allows for convenient and remote monitoring of attendance. Telegram messages provide real-time updates and notifications, enabling administrators or users to stay informed about attendance status.

REFERENCE:

- [1]. Sudheer B, S Rahul, S Manoj, Mrs. Vishalini Divakar, "Attendance System Using Face Recognition And Contactless Thermal Screening" Vol. 3, No. 5, Pp. 368-370, 2022.
- [2]. Smitha, Pavithra S Hegde, Afshin, "Face Recognition Based Attendance Management System", International Journal of Engineering Research & Technology (IJERT) Vol. 09, Issue. 05, 2020.
- [3]. Prof. Mohini Mohan Sawarkar¹, Mr. Atif Imran Penkar, "Face Recognition and Fingerprint based Attendance Monitoring System" Vol. 07, Issue. 03, 2020.
- [4]. Mekala V, Vibin Mammen Vinod, Manimegalai M, "Face Recognition Based Attendance System" ISSN: 2278-3075, Vol. 8 Issue. 12, 2019.
- [5]. Tippavajhala Sundar Srinivas, Thota Goutham, Dr. M. Senthil Kumaran, "Face Recognition based Smart Attendance System Using IoT", International Research Journal of Engineering and Technology (IRJET), Vol. 09, Issue. 03, 2022.
- [6]. Dr. Manjunath B E, Dr. S. Koteswara Rao, K. Rambabu, "Iot Based Smart Attendance Monitoring System Using Raspberry Pi-Based Face Recognition", Journal Of Cardiovascular Disease Research, Vol. 13, Issue. 06, 2022..
- [7]. D. Narendar Singh, M. Kusuma Sri, K. Mounika, "IOT Based Automated Attendance with Face Recognition System", International Journal of Innovative Technology and Exploring

- Engineering (IJITEE), Vol. 8, Issue. 6S4, 2019.
- [8]. Kumar, K.S., Perumal, S., Mohan, R. and Kalidoss, K., 2016. Numerical Analysis of Triple Concentric tube Heat Exchanger using Dimpled Tube Geometry. Asian Journal of Research in Social Sciences and Humanities, 6(8), pp.2078-2088.
- [9]. Perumal, S, Mohan, R., Sasidharan, S. and Venkatesh, K., 2017. Study On Concentric Tube Heat Exchanger With Different Nano Fluids For Enhancing The Heat Transfer: A Review., Imperial Journal of Interdisciplinary Research, Volume 3, Issue 9, ISSN 2454-1362. 682-688.
- [10]. K.Senthilkumar,S.Perumal, P.Palanisamy., 2014 Numerical study on a concentric tube heat exchanger using dimpled tubes with al₂o₃ nanofluid., Australian journal basic and applied sciences, 8 (7), 185-193.
- [11]. Perumal saravanan and Mohan raman., 2020 Experimental And Numerical Analysis Of Diesel Engine Exhaust Heat Recovery Using Triple Tube Heat Exchanger, Thermal Science, 2020:24; 525-531.
- [12]. Ravi Shankar Saxena, M. Sireesha, P. Deekshita, K.Murali Sairam, K. Shanmukha Rao, K. Achyuth, "Automatic Attendance Management System Using Face Recognition", International Journal of Research Publication and Reviews, Vol. 3, No. 11, PP. 844-847 2022.
- [13]. Prajakta Bagul, Darshan Khedkar, Vaibhav Jagtap, "Attendance Monitoring System With Face Recognition Using Iot", International Journal Of Advance Scientific Research And Engineering Trends, Vol. 6, Issue. 2, 2021..
- [14]. Sandesh Kulkarni, Minakshee Bagul, Akansha Dukare, Prof. Archana Gaikwad, "Face Recognition System Using IoT", IJAR CET Volume 6, Issue 11, November 2017, ISSN: 2278 – 1323.
- [15]. Ravi Kishore Kodali, Vishal Jain, Suvadeep Bose and Lakshmi Boppana, "IoT Based Smart Security and Home Automation System", IEEE 2016.
- [16]. Shaik Anwar, D. Kishore, "IoT based Home security system with alert and door access control using Smart Phone", December 2016, IJERT.