MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY

IT5611- EMBEDDED SYSTEMS AND INTERNET OF THINGS

PROJECT REPORT

FACE DETECTION BASED ATTENDANCE SYSTEM USING IOT

TEAM : 21

JAGADEESH R – 2021506314

SANTHOSHKUMAR L – 2021506310

VIGNESH S – 2021506123

ABSTRACT:

The proposal is mainly about the face detection-based attendance system which overcomes manual method of taking attendance, reduces the time to take attendance and avoids proxy. Unlike other Biometric attendance systems like fingerprint recognition, DNA matching, face recognition, Iris recognition and voice recognition, this face detection-based attendance system does not require any person to be in contact with the fingerprint reader module, which reduces a lot of time, improves security levels and low fraud rate.

This project involves the student, faculty or employeeattendance by face recognition. The face detection and recognition were carried out by using ESP32 cam module which captures the images of the available students in the classroom. Using this image, it recognizes and compares with the images in the database and marks their attendance

Keywords: **Face Recognition, Face Detection, ESP32 camera module.**

INTRODUCTION:

In today's fast-paced world, managing attendance efficiently is a crucial task for organizations, institutions, and businesses. Traditional attendance systems, such as manual registers or swipe cards, can be prone to errors and time-consuming. With advancements in technology, automated face detection attendance systems have emerged as a reliable and convenient solution.

Attendance management is an integral part of any institution, be it educational, corporate, or organizational, as it directly influences productivity, resource utilization, and decision-making. Traditional methods, such as manual roll-calling or sign-in sheets, are laborious, time-consuming, and susceptible to errors.

The advantages of employing a face detection-based attendance system are multifaceted. Firstly, it offers seamless and frictionless user experience, with individuals being marked present simply by standing in front of a camera, without the need for any physical interaction. Secondly, the system mitigates issues of "buddy punching" or proxy attendance, as it uniquely identifies individuals based on their facial features, ensuring accountability and data accuracy.

Implementing a face detection-based attendance system poses certain challenges. These may include concerns regarding privacy and data security, as facial information is sensitive and requires stringent protection. Additionally, factors like varying lighting conditions, occlusions, and pose variations may affect the system's performance, necessitating careful consideration during design and deployment.

The Face Detection Attendance System using ESP32 is a cutting-edge application of Internet of Things (IoT) and computer vision technologies. It combines the power of ESP32, a versatile microcontroller, and sophisticated facial recognition algorithms to create a seamless and accurate attendance management system.

LITERATURE SURVEY:

Taking attendance manually is a time-consuming process that frequently results in human mistakes. This paper describes the use of Open-Source Software to create a Student Attendance Management System in a multi-user setting. For reading tags, the system employs Python as a backend. To authenticate lecturers, run the Python code, and record tags in a sheets file. Finally, the sheet file is posted to the server, where it will be processed and interpreted. The web portal can be used to check attendance.

We have discovered that we have many technologies to use, but we are still following the traditional way. When it came to the attendance system in universities and schools, professors had to do everything by hand. Lecturers took attendance and manually updated it in the database. When it comes to technology, we've discovered that there are numerous instruments that may be used to ease the strain of lectures.(rewrite dont use we) One example of this is the use of IOT (Internet of Things). We can accomplish it automatically, eliminating the need for lectures. For greater speed, we intend to leverage the Cloud as storage, using the Internet of Things.

(include the citation)The Eigenfaces algorithm was developed by Sirovich and Kirby (2013) and used by Matthew Turk and Alex Pentland in face classification. This algorithm captures the variation in the collected faces and compares it with individual faces by generating a set of eigenvectors from the image by the principal component analysis. The eigenvectors are derived based on the covariance matrix of the probability distribution over the high-dimensional vector space of face images

Fisher faces algorithm is derived from an idea suggested by R.A.Fisher in 2014. This algorithm is an enhancement to Eigen faces that uses linear discriminant analysis. The linear discriminant analysis finds the linear combination of features that characterizes or separates two or more classes of faces. The resulting combination is used for the reduction of dimensionality. This algorithm is used when there is a large variation in the illumination and facial expressions

Local binary pattern histogram algorithm is described in the year 2016 and was a powerful technique for feature extraction. In this algorithm, the grayscale image is divided into cells of the 3X3 matrix. The obtained matrix is represented with the range of pixel intensities. The center value of the matrix is considered as the threshold and each pixel value is compared with the threshold. If the pixel value is greater than the threshold it is given 1 otherwise 0. An 8-digit binary value is converted into decimal value for convenience. Then the histogram is computed over the cells and frequency of each number and concatenate the histogram of all cells to obtain the feature vector. This feature vector is processed to classify the images

Scale-invariant feature transform algorithm was published by David Lowe in 2017. This algorithm detects and describes the local features in the images. Whenever the image is transformed, like the size of the image is altered or when it is rotated the features of the image are invariable. These invariable features are used to detect the image

Patil et al. in applied face recognition for classroom attendance in 2017. They used Eigenface for the recognition, but the overall accuracy of the system was not mentioned in the paper. Similar approach using Eigenface for the face recognition based attendance tracking system was proposed in. They achieved an overall recognition accuracy of 85% for unveiled faces.

Principal Component Analysis (PCA) method for the face recognition for their attendance system, achieving the accuracy of 68%.

Speed up robust feature algorithm was presented by Herbert Bay, Tuytelaars, T. and Van Gool in the year 2019. This algorithm is a patented local feature detector and descriptor. SURF descriptors are used to locate and recognize objects, people or faces, to reconstruct 3D scenes, to track objects and to extract points of interest. This algorithm uses a multiresolution pyramid technique where the image is transformed into coordinates to copy the original image with Pyramidal Gaussian shape to obtain an image with the same size but with decreased bandwidth

Many advanced algorithms have been developed in the last decade. But these algorithms demand more hardware requirements than the present hardware used to execute them without any difficulty.

The above complexities are overcome by a simplified face detection algorithm provided by ESP - 32 cam server and with a Arduino uno communication card which provides an easily accessible user interface for enrollment and detection of multiple faces.

PROJECT METHODOLOGY:

The project aims to develop a face detection attendance system utilizing the ESP32-CAM module as the core hardware platform. The methodology involves thorough planning and requirements gathering to define the system's scope and desired outcomes. The hardware setup entails assembling the ESP32-CAM module, USB to Serial TTL converter, power supply, SD card, and camera mounting.

The software setup involves configuring the development environment, including the Arduino IDE or PlatformIO, and incorporating essential libraries for ESP32, camera,and face detection, such as OpenCV. An appropriate face detection algorithm, like Haar cascadesor Single Shot Multibox Detector (SSD), is selected for efficient embedded performance. A datasetof faces is collected and labeled for training the face detection model, and the model is then exported to the ESP32-CAM.

The code is developed to capture video frames from the camera and apply the face detection algorithm in real-time, recording timestamps and individual IDs upon detection. Data storage and management systems are established, using an SD card to store face data and attendance records, organized in a well-structured format.

Optionally, face recognition can be implemented to further enhance accuracy and individual identification. Testing is conductedin diverse environmental conditions to optimize the system's accuracy and performance.

The ESP32-CAM system, along with any optional user interface, is deployed for attendance tracking, and comprehensive documentation is created to facilitate future maintenance and iterations. Regular monitoring and maintenance are performed to ensure the system's reliability and efficiency, providing an effective and automated face detection attendance solution.(explain the figure)

SCHEMATIC DIAGRAM:

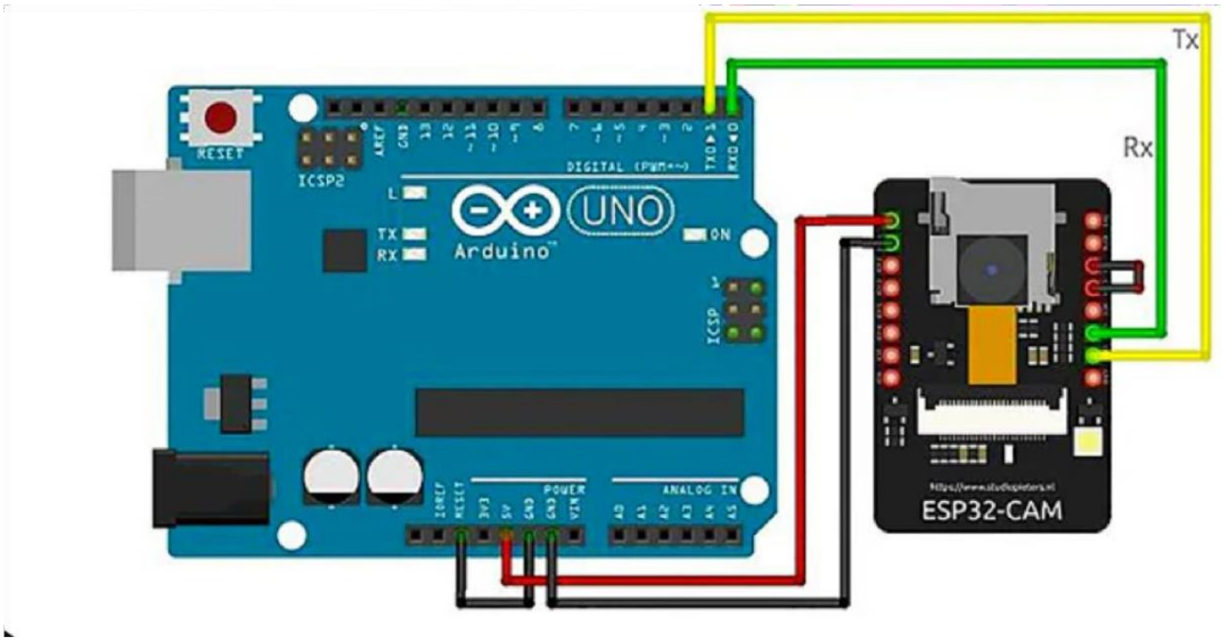


Fig 1: Face Detection Schematic Diagram

BLOCK DIAGRAM:

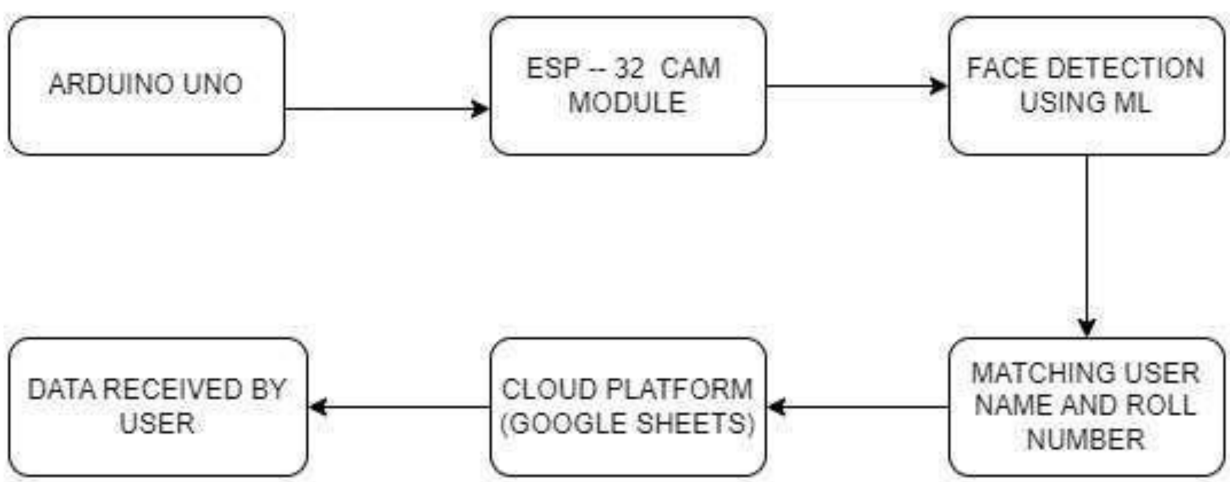


Fig 2: Face Detection Attendance System

COMMUNICATION CARD:

To ensure the uploading of an operating program of such an application to the ESP32-CAM card, there are several cards such as FTDI, ARDUINO, MODULE TTL. We chose the ARDUINO board as a converter compatible with the control board because it can play both a control board and a converter.

CAMERA: ESP - 32 CAM:

It is a relatively small integrated module consisting of an OV2640 camera and an ESP3-S processor that enables the creation of IP camera projects for video streaming with various resolutions and visualization in real time via a wifi network. It has three 3.3V or 5V GND power pins.

For night photos, for example, a high-brightness LED can be bought. In this situation, a photoresistor can be connected to the board. It does, in fact, provide some GPIO ports for connecting various sensors and actuators.

We use this camera because of :

We employ this camera because of its data transfer speed, storage capacity, efficient transmission without data loss or noise intrusion, effective receipt of transmitted signals (such as images; sensor responses), and responsiveness to order reception and execution.

The ESP-32CAM is suitable for a wide range of IoT applications. It is appropriate for smart home appliances, industrial wireless controls, wireless monitoring, QR wireless identification, wireless positioning system signals, and other Internet of Things (IoT) applications. It is an excellent choice for IoT applications.

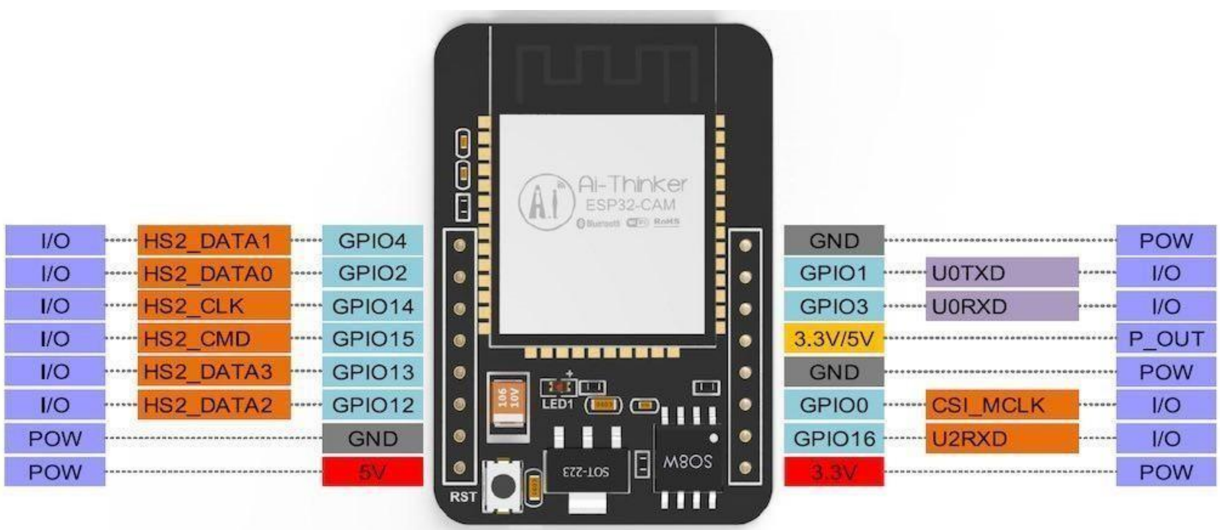


Fig. 3: Internal composition of the ESP32-CAM board

ALGORITHM :

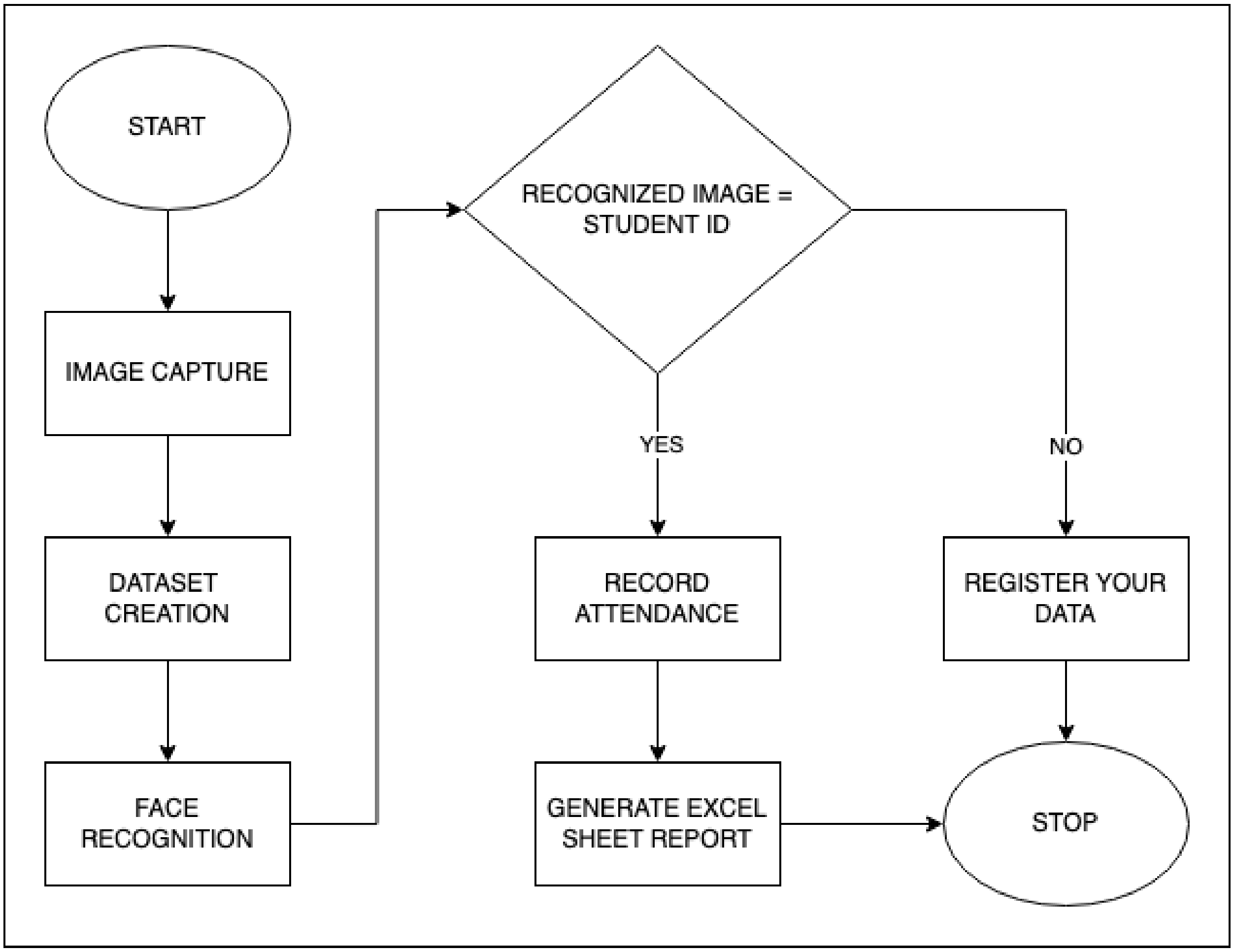


Fig. 3.1: Algorithm For Face Recognition Attendance system

SOURCE CODE :

```
#include "esp_camera.h"
#include <WiFi.h>

//          Ensure ESP32 Wrover Module or other board with PSRAM is selected
//          Partial images will be transmitted if image exceeds buffer size

// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
```

```
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
//#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
#include "camera_pins.h"

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

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

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

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  // for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }
```

```
#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

FACE DETECTION:

In this section, we will establish a network connection to the ESP32-CAM board. The router functions as an access point, while the ESP32-CAM board functions as a station. We connect to a local network to operate and access the ESP32-CAM web server.

The ESP32-Cam module's network access mechanism, which includes Wifi connectivity, enables it to send movies in real time. The module is essentially composed of a microcontroller that operates similarly to an Arduino, allowing it to be programmed using the Arduino software.

The code is developed to capture video frames from the camera and apply the face detection algorithm in real-time, recording timestamps and individual IDs upon detection. Data storage and management systems are established, using an SD card to store face data and attendance records, organized in a well-structured format.

To upload the program from the computer to the ESP32-Cam, the first step is to connect the module to a PC via a programmer. Hosted by a web server, the ESP32-Cam module can create its own Wifi network which allows it the autonomy to generate a connection and send videos to any location.

The face recognition algorithm by ESP32-CAM is given as follows:

Step 1: Access via web browser

The serial monitor displays the IP address that will allow the opening of the camera according to the URL reserved in a web browser

```
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.43.192' to connect
```

Fig. 4: Displaying data on the serial monitor

Copy the address given by Fig.8 and then paste it into the address bar of a web browser. The web browser needs to be on the same network as the ESP32-CAM is connected to.

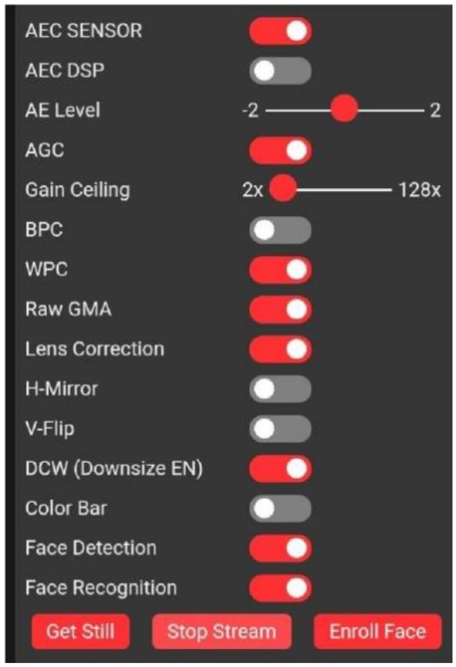


Fig. 5: Camera feature

Step 2 : Validation of the authentication phase

Each time the camera succeeds in detecting a face, an identification is made after a real-time comparison between the detected face and those registered in our database.

Once the comparison is made, the application can provide two results: the name of the person if the face is well selected in the database or an unknown person since his face does not exist in the database.

Demonstrate the proper operation of the phenomena of facial recognition by using numerous instances with grimaces and varied positions indicating the percentage of confirmation of the requested person.

SENDING ATTENDANCE DATA TO GOOGLE SHEETS:

As previously stated, attendance data from the ESP32-CAM will be provided to Google Sheets. We'll be using IFTTT applets for this. This applet will use webhooks to retrieve data from the ESP32-CAM and then update it in Google Sheets

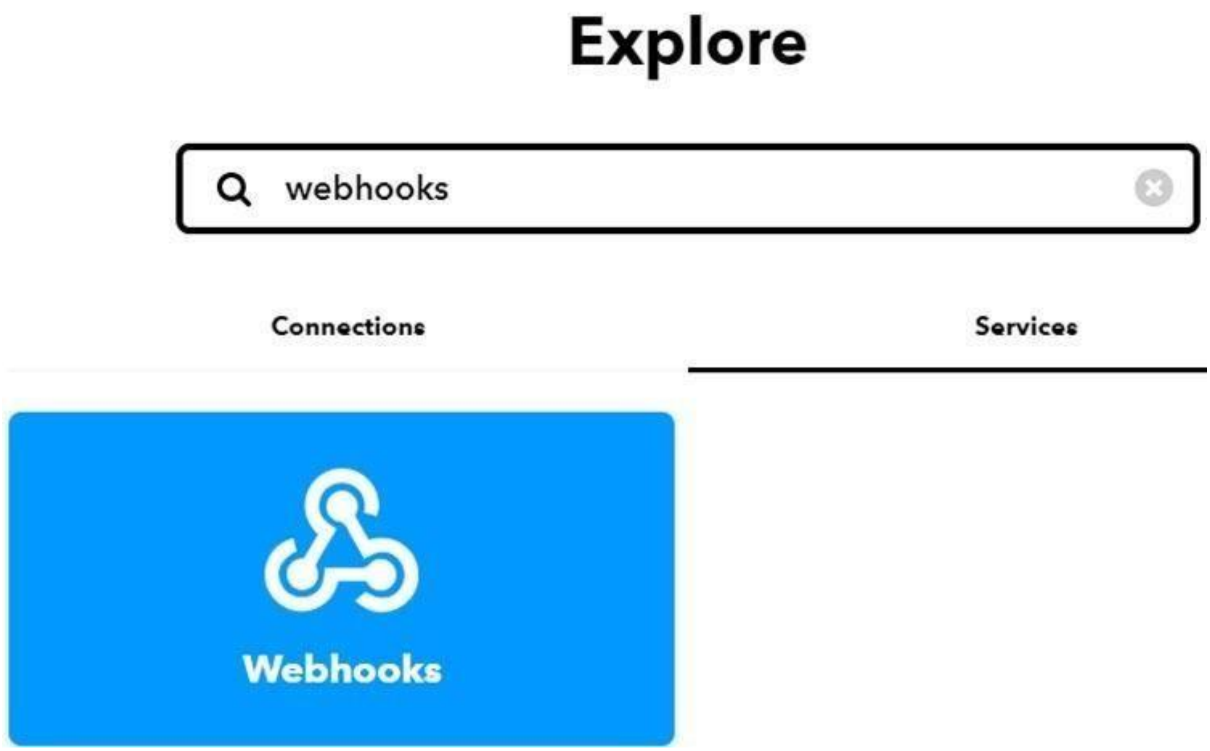


Fig 6: Choosing a service to receive web requests from esp-32 cam

When it gets the request from the webhooks it will initiate a request to google sheets and it will store the attendance in the excel sheets by adding rows to each person.

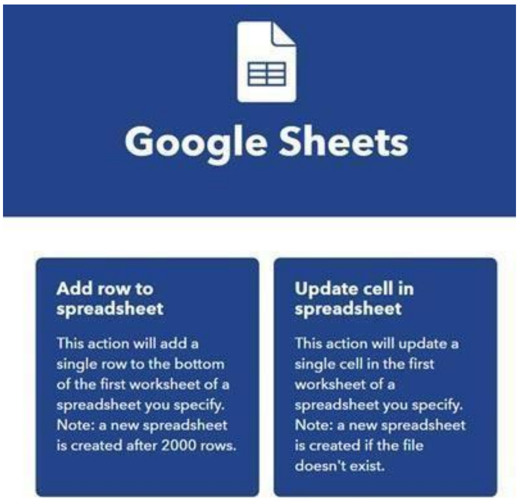


Fig 7:Google sheets for accessing attendance data

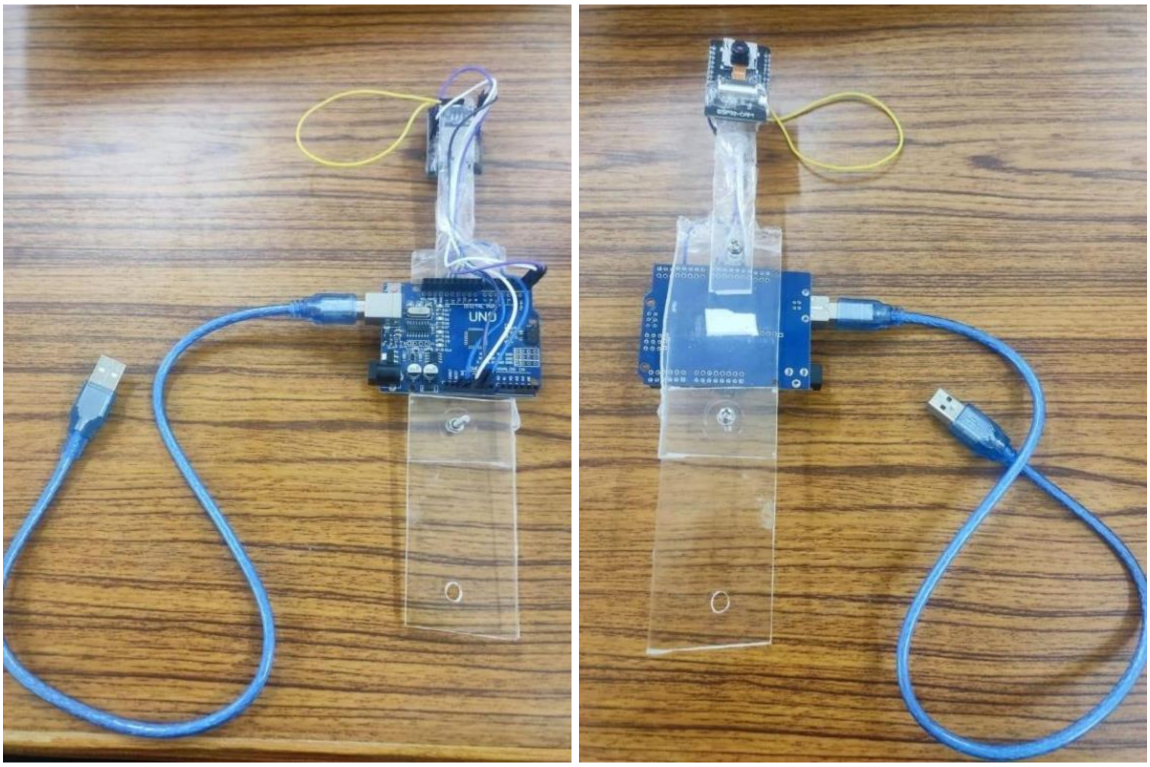


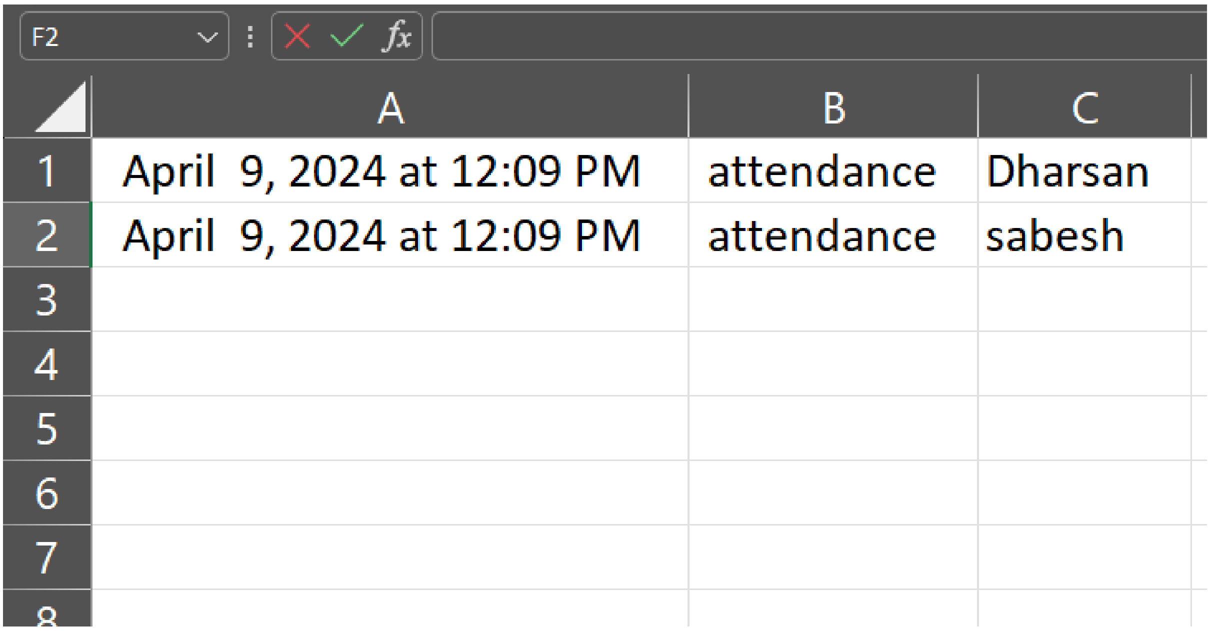
Fig 8: Project image

RESULTS AND DISCUSSIONS:

Each people must first register their faces. After each student has been enrolled, the camera will begin identifying faces in the frame whenever the student walks in front of it. After detecting, thesystem will automatically mark the attendance of the specific student in the attendance list in the Excel sheet of a certain topic with their name and time. Students' attendance is recorded in an excelsheet along with the time it was recorded.

Fig 9: Face Detection of Person 1

Fig 10: Face Detection of Person 2



	A	B	C
1	April 9, 2024 at 12:09 PM	attendance	Dharsan
2	April 9, 2024 at 12:09 PM	attendance	sabesh
3			
4			
5			
6			
7			

Fig 11 : Employee Attendance Sheet

CONCLUSION:

Different classrooms may have different lighting, seating layouts, and atmosphere. There may also be students with diverse facial expressions, hair styles, beards, spectacles, and so on. This approachcan be adopted to improve the management of attendance and leaves. The system will save time, reduce the amount of work that the administration needs to do, replace stationery with electronic equipment, and minimize the quantity of human resource required for the purpose.

REFERENCES:

1. Mahesh Sutar, Mahesh Patil, Sachin Waghmare, “Smart Attendance System Using RFID In IOT”, International Journal of Advanced Research in Computer Engineering &Technology (IJARCET)
2. T. Lim, S. Sim, and M. Mansor,” RFID based attendance system”, in Industrial Electronics and Applications, 2009. ISIEA 2009. IEEE Symposium on, vol. 2. IEEE, 2009, pp.778782.
3. Beymer, D. and Poggio, T. (1995) Face Recognition from One Example View, A.I.Memo No. 1536, C.B.C.L. Paper No. 121. MIT
4. Viola, Paul, and Michael J. Jones, “Robust real-time face detection”, Internationaljournal of computer vision 57.2(2004):137-154.
5. R. Girshick, “Fast R-CNN,” in IEEE International Conference on Computer Vision(ICCV), 2015
6. Yang B, Yan J, Lei Z, et al. Convolutional Channel Features for Pedestrian, Face and Edge Detection [J]. Computer Science, 2015:82-90.
7. Michał Dolecki, Paweł Karczmarek, Adam Kiersztyn, Witold Pedrycz, "Face recognition by humans performed on basis of linguistic descriptors and neural networks", Neural Networks (IJCNN) 2016 International Joint Conference on, pp.5135-5140, 2016, ISSN 2161-4407
8. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deepconvolutional neural networks,” in Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held on December 3-6,2012, (Lake Tahoe, Nevada, United States), pp. 1106–1114, 2012
9. Soniya, Paul S, Singh L. A review on advances in deep learning[C]// IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions. IEEE, 2015:1-6.
10. R. Sokullu, O. Akin, and E. Aslan, "Face Recognition Based Multifunction Smart Mailbox," 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), pp. 1-4, (2020),
11. Ming Du, Aswin C. Sankaranarayanan, and Rama Chellappa, Robust Face Recognition from Multi-View Videos, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 3, MARCH 2014.