

# Face Recognition Based Attendance System Using ESP32-Cam

A. Ajay, K. Shashikanth, B. Yeshwanth reddy

Department of Electronics & Communication Engineering

ACE Engineering College, Hyderabad, India

**Abstract:** *The Face Recognition Based Attendance System using ESP32-CAM is an IoT-based project designed to automate the process of attendance marking using facial recognition technology. Traditional attendance systems, such as manual entry or ID-based authentication, are time consuming and prone to proxy attendance or errors. To overcome these limitations, this project employs the ESP32-CAM module, which is a low-cost microcontroller integrated with a camera and Wi-Fi capabilities, making it ideal for real-time image processing and cloud communication. In this system, the ESP32-CAM captures the images of individuals in front of the camera and performs face detection and recognition using pretrained machine learning models. Once a face is identified, the system automatically records the person's name, ID, and timestamp into a database (such as Firebase or Google Sheets) through Wi-Fi connectivity. The recognized faces are compared against the stored dataset, and attendance is updated instantly without human intervention. This project demonstrates the integration of IoT, computer vision, and artificial intelligence for efficient attendance management. It offers a cost effective, contactless, and scalable solution suitable for schools, offices, and organizations. The proposed system not only enhances accuracy and transparency but also reduces the administrative workload associated with conventional attendance methods.*

**Keywords:** *Face Recognition Based Attendance System*

## I. INTRODUCTION

Attendance management is an essential task in schools, colleges, and organizations. Traditionally, attendance is recorded manually using registers or sheets. However, this method consumes time, is prone to human errors, and allows proxy attendance. To overcome these problems, automated attendance systems are developed using modern technologies such as biometrics, RFID, and face recognition. Face recognition is a biometric technology that identifies a person based on facial features. Unlike fingerprint or card-based systems, face recognition does not require physical contact and provides a convenient method of authentication. It works by capturing an image of a person's face and comparing it with stored facial data in a database.

The Face Recognition Based Attendance System using ESP32-CAM is an automated system that uses a camera to capture the face of a person and mark attendance automatically. The ESP32-CAM module contains a camera and Wi-Fi capability, allowing it to capture images and send data to a server or database. When a registered person's face is detected and recognized, the system automatically records attendance. The system operates by first enrolling the faces of students or employees in the database. When a person appears in front of the camera, the system detects the face and compares it with the stored images. If the face matches with a registered user, the attendance is marked and stored in the database.

This system reduces manual work, increases accuracy, and prevents proxy attendance. It is widely used in educational institutions, offices, and industries for attendance monitoring. Modern microcontrollers like the ESP32-CAM enable low-cost and compact implementations of face recognition systems because they integrate Wi-Fi connectivity and camera support in a single module. The system can capture images, process facial features, and transmit attendance data wirelessly



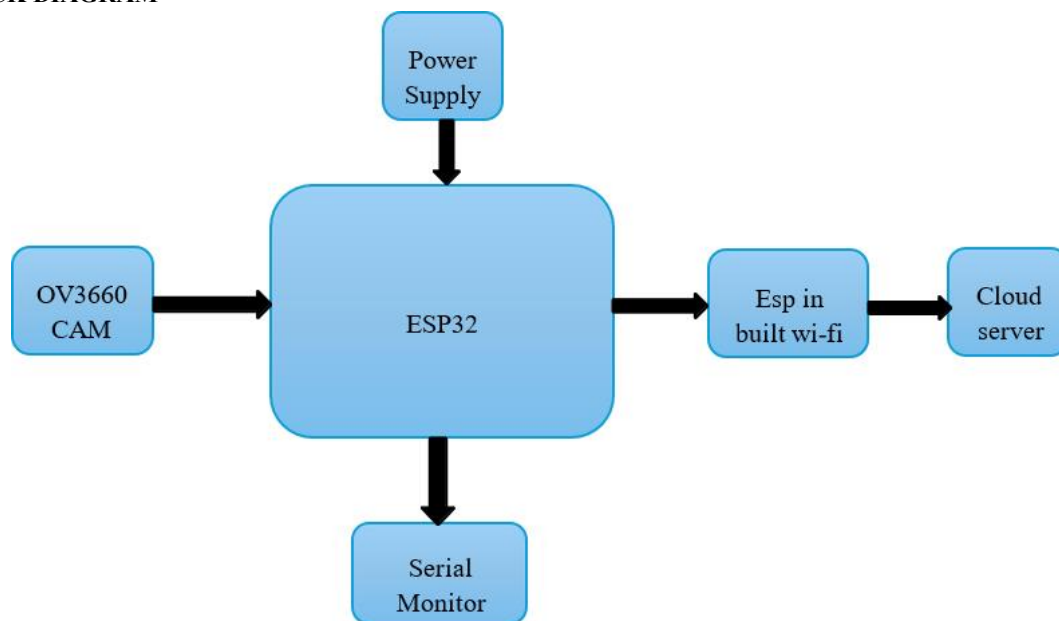
## II. RELATED WORK

Face recognition technology has been extensively researched in the fields of computer vision and artificial intelligence. Early systems relied on basic image processing techniques such as edge detection and template matching. However, these methods had limitations in accuracy and robustness.

Later developments introduced advanced algorithms such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Eigenfaces for improved recognition performance. With the rise of machine learning and deep learning, Convolutional Neural Networks (CNNs) became widely used for face detection and recognition due to their high accuracy.

Recent studies have focused on implementing face recognition systems using embedded devices such as Raspberry Pi and ESP32 modules. These systems provide cost-effective and portable solutions suitable for real-time applications.

### BLOCK DIAGRAM



## III. SYSTEM ARCHITECTURE AND HARDWARE DESIGN

### 1. System Architecture

The system architecture of the face recognition attendance system consists of image capture, face detection, face recognition, and attendance marking.

The ESP32-CAM module captures real-time images and processes them to detect faces. Once a face is detected, it is compared with pre-stored facial data. If a match is found, the system marks attendance and stores the data.

The system can also send attendance data to a server or display it via a web interface using Wi-Fi connectivity.

Main Stages in System Architecture:

1. Image Capture using ESP32-CAM
2. Face Detection
3. Face Recognition
4. Attendance Marking
5. Data Storage and Monitoring



## 2. Hardware Design

### 2.1 ESP32-CAM Module

The ESP32-CAM is the main component of the system. It includes a microcontroller, camera, and Wi-Fi module. It is responsible for capturing images and processing face recognition algorithms.

### 2.2 Camera (OV2640)

The camera module captures real-time images of individuals. It provides sufficient resolution for accurate face detection and recognition.

### 2.3 Power Supply

A regulated power supply is used to provide stable voltage to the ESP32-CAM module.

### 2.4 Storage System

Facial data is stored in the memory for comparison during recognition. Attendance records can also be stored locally or on a server.

## IV. SOFTWARE DESIGN AND SYSTEM OPERATION

### 1. Software Design

The software design involves programming the ESP32-CAM using Arduino IDE or similar platforms. Image processing and face recognition algorithms are implemented within the system.

The system uses libraries for face detection and recognition, enabling efficient processing of captured images.

Key Functions of the Software:

- Image capture and processing
- Face detection using algorithms
- Face recognition using stored data
- Attendance recording
- Wi-Fi communication for data transfer

### 2. System Operation

The system begins by initializing the ESP32-CAM and connecting to Wi-Fi. The camera captures images continuously. When a face is detected, the system compares it with stored facial data. If a match is found, the attendance is marked with a timestamp.

If no match is found, the system ignores the face or can store it for future registration.

### 3. Operational Steps

1. Initialize ESP32-CAM and connect to Wi-Fi
2. Capture image using camera
3. Detect face in the image
4. Compare with stored database
5. Mark attendance if matched
6. Store attendance data
7. Repeat process continuously



**IV. SOURCE CODE**

```
#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <HTTPClient.h>
#include <WebServer.h>
//

// ===== ESP32-CAM (AI Thinker Layout Pins) =====

// DO NOT CHANGE THESE

//

#define PWDN_GPIO_NUM 32

#define RESET_GPIO_NUM -1

#define XCLK_GPIO_NUM 0

#define SIOD_GPIO_NUM 26

#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35

#define Y8_GPIO_NUM 34

#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5

#define VSYNC_GPIO_NUM 25

#define HREF_GPIO_NUM 23

#define PCLK_GPIO_NUM 22

// =====

// WIFI
```



```
// =====  
  
const char* ssid = "OnePlus 11R 5G"; const char* password = "sonu1234";  
// =====  
  
// BACKEND  
  
// =====  
  
const char* backendBaseUrl = "http://10.119.70.74:5000";  
  
// Optional buzzer pin (-1 = disabled) #define BUZZER_PIN -1  
  
WebServer server(80);  
  
unsigned long lastCaptureMs = 0;  
  
const uint32_t captureIntervalMs = 5000;  
  
// =====  
  
// CAMERA INIT  
  
// =====  
  
bool initCamera() { 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;  
//  IMPORTANT for OV3660 stability  
  
config.frame_size = FRAMESIZE_QVGA; // 320x240 config.jpeg_quality = 15; // Stable config.fb_count = 1; //  
Prevent overflow  
if (psramFound()) { config.fb_count = 2;  
}  
  
esp_err_t err = esp_camera_init(&config); if (err != ESP_OK) {  
Serial.printf("Camera init failed: 0x%x\n", err); return false;  
}
```



```
sensor_t * s = esp_camera_sensor_get();

if (s) {

s->set_framesize(s, FRAMESIZE_QVGA); s->set_quality(s, 15);
}

Serial.println("Camera initialized successfully ☐"); return true;
}

// =====

// WIFI CONNECT

// =====

void connectToWiFi() { WiFi.mode(WIFI_STA); WiFi.setSleep(false); WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");

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

}

Serial.println("\nWiFi connected ☐"); Serial.print("IP Address: ");
Serial.println(WiFi.localIP());

}

// =====
// CAPTURE & UPLOAD
// =====

void captureAndSend() {
camera_fb_t * fb = esp_camera_fb_get();

if (!fb) {

Serial.println("Camera capture failed"); return;
}

WiFiClient client;

HTTPClient http;
```



```
String url = String(backendBaseUrl) + "/upload";
http.begin(client, url);
http.addHeader("Content-Type", "image/jpeg");
int httpCode = http.POST(fb->buf, fb->len);

if (httpCode > 0) {

Serial.printf("Upload OK, code: %d\n", httpCode);

} else {

Serial.printf("Upload failed: %d\n", httpCode);

}
http.end(); esp_camera_fb_return(fb);
}
// =====

// SETUP

// =====

void setup() {

Serial.begin(115200);

delay(1000);

Serial.println("\nBooting SmartVision ESP32-CAM...");

if (!initCamera()) {

Serial.println("Camera failed! Restarting..."); delay(3000);
ESP.restart();

}

connectToWiFi();

}
// =====

// LOOP
```



```
// =====  
void loop() {  
  if (WiFi.status() == WL_CONNECTED) {  
    if (millis() - lastCaptureMs > captureIntervalMs) { lastCaptureMs = millis();  
      captureAndSend();  
    }  
  }  
  delay(50);  
}
```

## V. DISCUSSION

The implemented system demonstrates efficient and accurate face recognition for attendance management. The ESP32-CAM module successfully captures and processes images in real time.

The system shows high accuracy in recognizing registered users while preventing proxy attendance. The contactless nature improves hygiene and usability.

However, performance may vary under different lighting conditions or angles. Optimization of image processing algorithms can further improve accuracy and speed.

Overall, the system proves to be a reliable and cost-effective solution for automated attendance tracking.

## VI. CONCLUSION

This project successfully designed and implemented a Face Recognition Based Attendance System using ESP32-CAM. The system automates attendance marking using facial recognition technology, eliminating manual errors and improving efficiency.

The ESP32-CAM provides a compact and affordable solution for real-time face recognition applications. The system demonstrates accurate performance, ease of use, and scalability for various environments.

Future improvements may include cloud integration, mobile application support, and advanced deep learning algorithms for enhanced accuracy under challenging conditions.

The proposed system is highly suitable for modern smart environments such as educational institutions, offices, and secure access control systems.

## ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide and faculty members of the Department of Electronics and Communication Engineering for their valuable support and guidance. We also thank our institution and family members for their encouragement throughout the project.

## REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Pearson Education, 2008.
- [2] W. Zhao et al., "Face Recognition: A Literature Survey," ACM Computing Surveys, 2003.
- [3] ESP32-CAM Technical Reference Manual, Espressif Systems.
- [4] S. Z. Li and A. K. Jain, Handbook of Face Recognition, Springer, 2011.

