**National University of Computer and Emerging Sciences**



**IoT-Based Face Recognition Smart Gate Control System**

**Course Project**

**Team members:**

M. Rafay Ul Haq 21L-5383

Abdul Rafay 21L-1786

Eisha Fayyaz 21L-1752

**Introduction**

In this project, we aim to develop a prototype IoT system that enables intelligent gate access control using face detection technology. The system uses an ESP32-CAM module to capture image frames, which are sent via MQTT protocol to a cloud-based AI model that performs face detection. If a face is detected, the system triggers an Arduino-controlled servo motor to open the gate automatically. This eliminates the need for RFID cards or manual gate operation.

**Project Aims**

The primary objective of this project is to develop a cost-effective and scalable IoT-based smart gate control system that utilizes face detection technology. The system aims to:

* **Automate Gate Access:** Detect human faces to automatically open the gate without manual intervention.
* **Enhance Security:** Ensure that only the presence of a human face triggers the gate mechanism, reducing unauthorized access.
* **Provide Real-Time Monitoring:** Offer a live video feed through a web-based dashboard for monitoring purposes.
* **Enable Remote Control:** Allow users to control the camera flash remotely via the web interface.

**Components**

The components include:

**ESP32-CAM:** Captures image frames and publishes them to the MQTT broker.

**MQTT Broker (flespi.io):** Facilitates communication between devices by handling topic-based message distribution.

**Cloud Backend:** Processes incoming image frames to detect faces using OpenCV's Haar Cascade classifier.

**ESP32 Devkit:** Subscribes to detection status and controls the servo motor to open/close the gate accordingly.

**Web Frontend:** Displays live video feed and provides controls for the camera flash.

**Servo Motor:** To open or close gate at 0 and 90 degrees.

**FTDI:** To program that ESP32 Cam board.

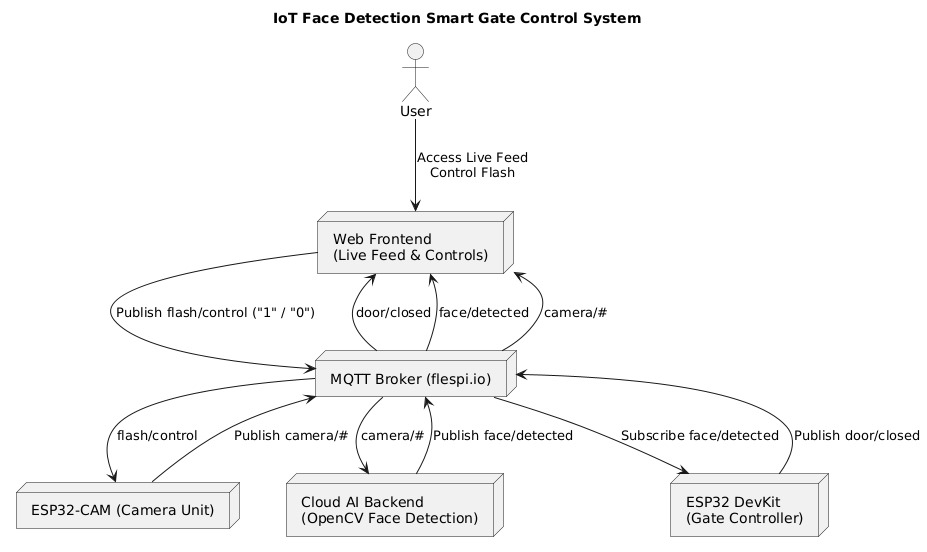
****

Fig 1. Block Diagram

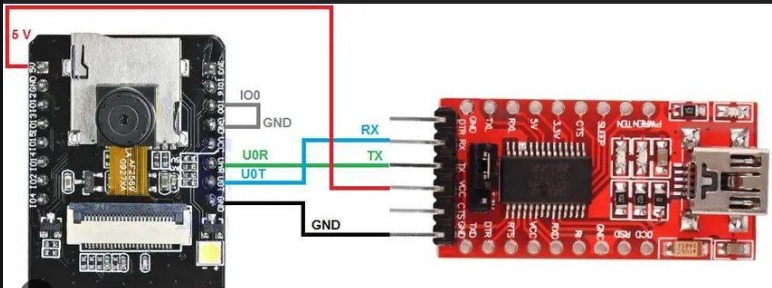
****

Fig 2. ESP32-CAM

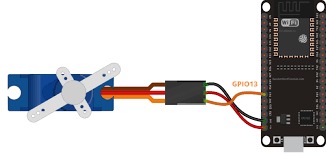
****

Fig 3. ESP32-Devkit

**MQTT Topics**

The system utilizes the following MQTT topics for communication:

**Camera to Cloud Backend:**

* camera/status: Publishes the status of the camera.
* camera/metadata: Contains metadata about the image frame (e.g., size).
* camera/frame/end: Indicates the end of an image frame transmission.
* camera/chunk/<index>: Transmits chunks of the image frame.

**Cloud Backend to Devices:**

* face/detected: Published as "true" when a face is detected in the frame.

**ESP32 Devkit to Devices:**

* door/closed: Published as "true" 10 seconds after the gate has been opened.

**Web Frontend to ESP32-CAM:**

* flash/control: Sends "1" to turn on the flash and "0" to turn it off.

**AI Model Details**

The cloud backend employs OpenCV's Haar Cascade classifier for face detection. This method is chosen for its efficiency and suitability for real-time applications on resource-constrained devices.

**Model Specifications:**

Classifier Used: haarcascade\_frontalface\_default.xml

**Detection Parameters:**

scaleFactor: 1.1

minNeighbors: 4

minSize: (30, 30)

**Performance Metrics:**

Accuracy: Approximately 95% under optimal conditions.

**Advantages:**

* Lightweight and fast, suitable for real-time detection.
* Does not require extensive computational resources.

**Limitations:**

* Prone to false positives and negatives, especially under varying lighting conditions and complex backgrounds.

**User Guide for Front-End**

The web-based frontend provides users with real-time monitoring and control capabilities.

**Features:**

* Live Video Feed: Displays the real-time video stream from the ESP32-CAM.
* Flash Control: Allows users to toggle the camera flash on or off.

**Gate Status Indicators:**

* Door Opened: Displays when a face is detected, and the gate is opened.
* Door Closed: Displays 10 seconds after the gate has been closed.

**User Interactions:**

Turning Flash On:

* Click the "Flash On" button.
* This action publishes "1" to the flash/control topic.

Turning Flash Off:

* Click the "Flash Off" button.
* This action publishes "0" to the flash/control topic.

**System Responses:**

* Upon detecting a face, the backend publishes "true" to the face/detected topic.
* The ESP32 Devkit, upon receiving this message, activates the servo motor to open the gate.
* After 10 seconds, the ESP32 Devkit publishes "true" to the door/closed topic, indicating the gate has been closed.