SCHOOL OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND ENGINEERING

UNIVERSITY OF NOTTINGHAM MALAYSIA

Designing intelligent agent Report

Designing Intelligent Agents (COMP3071 UNMC) (SPM1 22-23)

A face recognition-based door lock mechanism integrated with an attendance tracking system using Arduino and ESP32 cam.

Name	: Ting Chung Cheng
StudentID	: 20198633
Supervisor	: Dr. Nabil El Ioini & Dr. Rodney Petrus Balandong
Year	: 2023

Abstract

This paper presents the development of a face recognition-based door lock mechanism integrated with an attendance tracking system using Arduino and ESP32-CAM. The proposed system is designed to allow access only to authorized persons whose facial features have been previously captured and enrolled in the system. Besides, the system can track the attendance of individuals who enter or leave by sending name, date and time to a prepared google spreadsheet, this provides a convenient and reliable way to keep track of user attendance. The system utilizes the ESP32-CAM module to capture facial images, and the Arduino board to control the lock mechanism and store the attendance data by using self-created partition file. In summary, the proposed system provides a feasible and effective solution for both door security and attendance tracking, the proposed system are trusted to be applicable for certain situations and environments such as offices, schools, and other establishments that require high levels of security and attendance tracking.

Table of Contents

1. Introduction

The emergence of the Internet of Things (IoT) has brought significant advancements in the field of home automation, providing new possibilities to enhance the security and convenience of facilities. One of the most popular applications of IoT is smart security systems. Smart security systems mostly included smart locks, smart doorbells and etc which can be controlled and monitored remotely using certain connected sensors such as camera.

A smart door locking system usually plays a significant role in a smart security system, and facial recognition technology that uses biometric data can provide a secure and convenient way to authenticate the user's identity. In this context, the Arduino and ESP32-CAM, two popular microcontroller board be used to create an IoT-based smart door locking system with facial recognition.

The system is built on a combination of hardware and software components. The Arduino board provides the control and processing power needed to operate the system, while the ESP32-CAM module provides the camera and network connectivity needed for facial recognition. The software components of the system are written in Arduino IDE and C++ programming languages been used in this smart home automation approach. Besides, Google Sheet was involved in this project to provide attendance tracking feature.

In the following sections, components used in this system, the system's design and implementation, the algorithm used, the wiring connection and its potential applications in various industries will be discussed. Additionally, the attendance data tracked by the system will be automatically sent to a designated Google Sheet, the details will be explained in further section.

1.1 Aims

The final aim for this project is to build a Face recognition-based door lock mechanism. The agent was expected to build and work on Arduino uno microcontroller with certain module such as ESP32 cam, relay, solenoid door lock and etc.

The project aims are stated as follows:

- Evaluate existing home automation smart locking systems
- Build a face recognition system with door lock mechanisms
- Build an attendance tracking system
- Build a real-life hardware prototype

1. 2 Project motivation

The motivation behind developing an IoT-based smart door facial recognition locking system with Arduino and ESP32-CAM is to provide a high level of security and convenience to homeowners and businesses. The traditional lock and key system can be easily breached, and lost keys can lead to unauthorized access. In the other hand, a facial recognition locking system provides a more secure and reliable way to control access to a building or home. Besides, this system is trusted could eliminates the need for physical keys as well, since physical keys including key cards can be easily lost or duplicated.

The development of an IoT-based smart door facial recognition locking system with Arduino and ESP32-CAM presents an opportunity for me to deepen my understanding of electrical and electronic engineering, as it primarily falls under this field of study.

2. Background

Chapter 2 of this article details the project's background, which mainly focus on home automation. Section 2.1 will briefly describe about Arduino; a microcontroller been used in this project. While section 2.2 will go through ESP32 cam, a main sensor to collect image data and provide WIFI network connection. Section 2.3 provides an overview of face detection and recognition.

2.1 Arduino

Arduino is an open-source electronics platform designed for building and prototyping electronic devices. Multiple microcontroller boards (Gupta, 2023) such as Arduino UNO, Arduino Lilypad, Arduino Mega and etc provides a seamless electronic development environment. Arduino UNO R3, the Arduino board been used in this approach implementation is a microcontroller based on the ATmega328 associated with an open-source software (Watelectronics, 2021). It provides 6 analogue pins, 3 ground pin, 2 power output pin (3.3v and 5v) and 14 digital I/O pins that allows additional sensors and modules to be connected to the board externally. The Arduino UNO R3 pinout diagram is illustrated as follows in Figure 1 (Team Arduino, 2023).

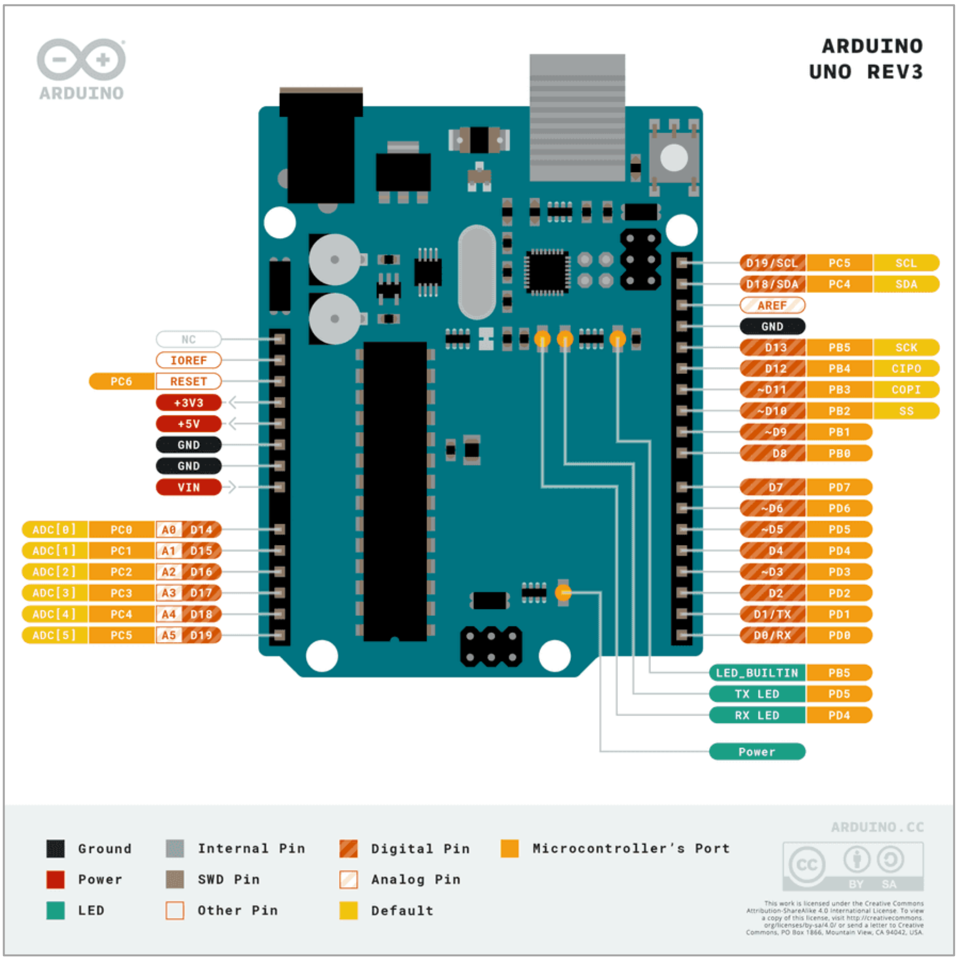


Figure 1: Arduino UNO R3 pinout diagram

Furthermore, a table depicting the various pins on Arduino UNO R3 and their corresponding functions will be presented in Figure 2.

Pin	Details
Vin	Power supply input to Arduino board when connects to external power supply
3.3V, 5V	Power supply pin which to provides certain voltage
GND	Ground pins
Reset	Reset and reboot the microcontroller when connected
A0 - A5	Analog input provides in range 0-5V
Digital Pins 0 -13	Input or output digital pins
0(Rx), 1(Tx)	TTL serial data receive and transmit
2, 3	Interrupt trigger
3, 5, 6, 9, 11	8-bit Pulse Width Modulation (PWM) output
*Board SPI pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Synchronous serial communication interface specification used for short-distance communication.
13	Built-in LED trigger
A4 (SDA), A5 (SCA)	Two-wired communication
AREF	Reference voltage provider

Figure 2: Arduino Pin table and their uses

As mentioned previously that Arduino is an open-source electronics platform designed for building and prototyping electronic devices. It is comprised of a physical programmable circuit board and a software development environment used to write and upload code to the board. The Arduino language used in Arduino IDE is mainly on C++, but it has its own unique varieties if compared to C++ language. The Arduino language has a lot of perception built in features, especially in the hardware interfaces, libraries, board manager, which could help developers pick it up easily. This allows Arduino became one of the famous and popular platforms by creating some attractive projects that can be used in a variety of applications, including robotics, home automation, and IoT devices.

2.2 ESP32 cam

ESP32-CAM, a compact camera module with dimensions of only 27*40.5*4.5mm and a deep sleep current as low as 6mA that released by Essence company. (Technology, A.-T., 2018). The ESP32-CAM can be used in a widely range of IoT applications such as industrial wireless control, wireless monitoring, wireless identification using QR codes, wireless positioning systems, and home automation. One of the important things to note is that ESP32 cam has a built-in Wi-Fi and Bluetooth module that enables it to connect to the internet and interact with other devices. This allows the cam could communicate with other devices and sensors by sending wireless signals and packets. In addition, this feature enables developers to remotely access the videos and image captured in real-time using the same hosting address. Figure 3 (JustAuser et al., 2020) illustrates the pin out diagram of ESP32 cam.

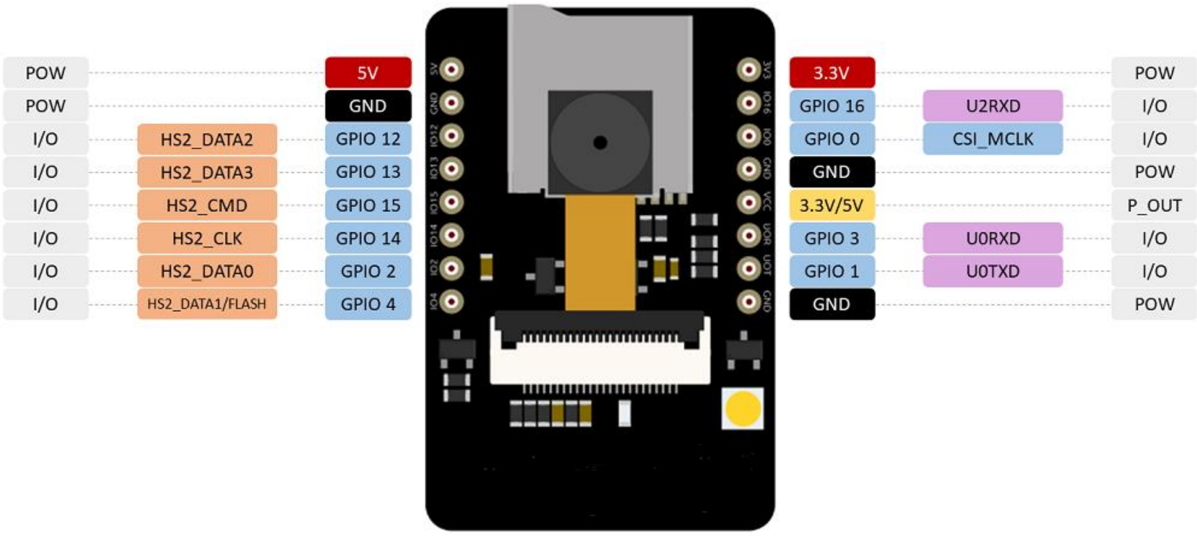


Figure 3: ESP32 cam pin out diagram

Besides, Figure 4 illustrates the table that details the pins and their relevant uses.

Pin	Details
3.3V, 5V	Both 3.3V and 5V power pin could be used to power the cam.
GND	Ground pin
3.3V/5V	Output power pin
GPIO 1 and GPIO 3	Serial Pins, (TX and RX, respectively)
GPIO 0	Determine whether the cam in flashing mode
GPIO 2, GPIO 4, GPIO 12, GPIO 13, GPIO 14, GPIO 15	Regular inputs/outputs. (When no SD card inserted)

Figure 4: ESP32 Cam pin table

2.3 Face detection and recognition

According to (Identification, 2022), face recognition is a biometric technology that involves identifying or verifying the identity of an individual based on their facial features. It is one of the most popular and widely used biometric identification techniques among others such as fingerprint, iris recognition and etc due to its non-intrusive nature and ease of use. According to (Patil1, 2013), facial features scored the highest compatibility in a Machine

Readable Travel Documents (MRTD) system, such as enrolment, security system, machine requirements, renewal, surveillance system and public perception, which as shown as in figure 5.

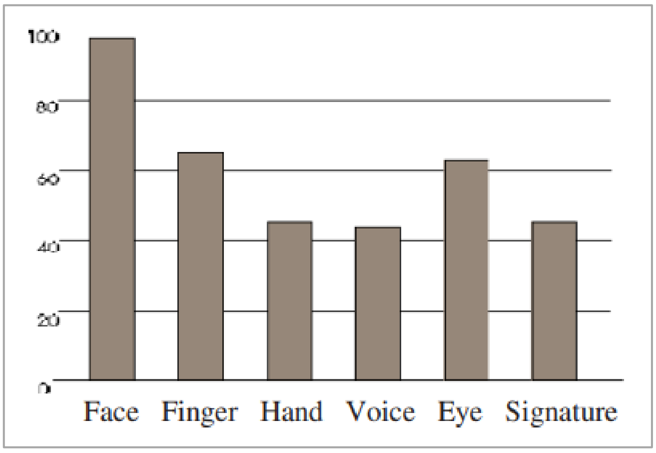


Figure 5: A comparison of various biometric features based on MRTD compatibility

The technology could be used in various applications, including security systems, access control, surveillance, and authentication.

There are total 5 different types of facial recognition technology till today (Felix, 2023), which are: Holistic matching method, Feature-based method, Hybrid types, Skin texture analysis and Thermal cameras.

The face recognition process is more or less the same according to (Dulčić, 2020), the figure 6 below shows an example of face recognition pipeline.

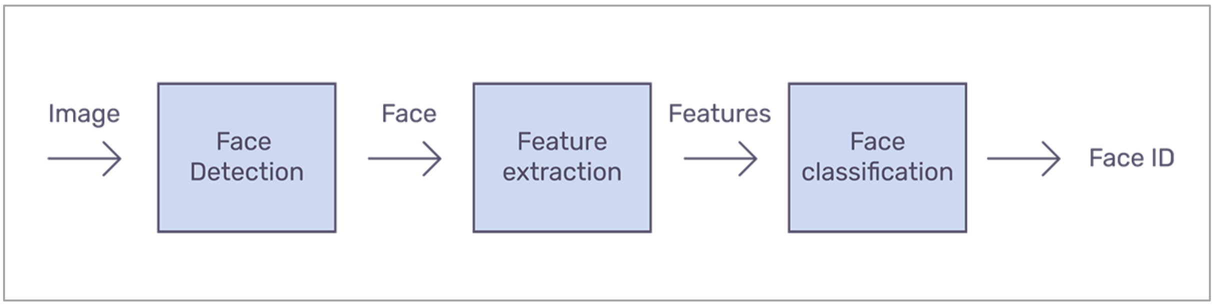


Figure 6: Face recognition pipeline (Dulčić, 2020)

The first stage, face detection, is the process of identifying one or more faces within an image or a video stream. Face detection typically uses algorithms that can operate at various levels of complexity, from simple template matching to more advanced techniques such as Haar cascades or deep learning-based approaches.

Feature extraction stage involves encoding the most important features of the face. This usually could be done by getting some specific details such as the distance between the eyes, the shape of the face and etc. Feature extraction could help creating a unique representation of each face so that the data comparing process will be much more reliable.

Final stage, classification stage is to compare the extracted features of faces with collected faces. This is typically done using a distance metric, such as Euclidean distance or cosine similarity, which measures the similarity between two face embeddings. The system will classify the face and match the comparison once the metrics reach certain thresholds.

According to (Allan, 2023), MTCNN was utilized for face detection, which can identify and locate faces in an image. On the other hand, face recognition, which is the process of identifying a specific individual's face, is performed with FaceNet.

Although face recognition technology has undergone significant advancements over the years recently, with increased accuracy and speed. However, this technology still faces several challenges such as variations in lighting conditions, facial expressions, and occlusions such as masks or scarves. Researchers are continuously working on developing more robust and accurate algorithms to improve the technology's performance and overcome these challenges.

2.3.1 MTCNN

MTCNN, the full name stands for Multi-task Cascaded Convolutional Network, which is one of the most popular and most accurate face detection tools that included in ESP32 cam face recognition process. There are 3 neural networks (P-Net, R-Net, and O-Net) been consisting of and connected in a cascade. (Dulčić, 2020) (Wang, 2018). Figure 7 below illustrated the structure of MTCNN model.

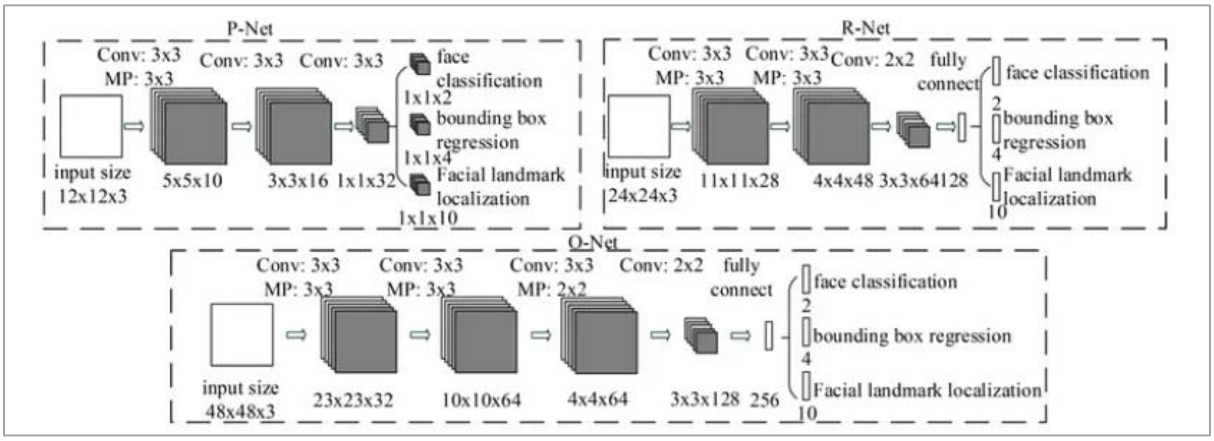


Figure 7: MTCNN structure

According to Figure 7 and (Wang, 2018), MTCNN could be differ into 3 different stages, while the 1st stage is to pass an image to the program. Different copies of the same image in different sizes were expected been created to search for different sized faces within the image (Zhang et al., 2016).

Multiple scaled copies of the image will be pass into the first neural net — P-Net — and its output will be gathered. After that, a relatively accurate bounding box for every 12 x 12 kernel will be create and been pass to stage 2.

2nd stage, R-net is similar to P-net but includes the coordinates of the new, more accurate bounding boxes, as well as the confidence level of each of these bounding boxes. Besides, the coordinates of the bounding boxes will be standardized, reshaped to square and pass to the last stage.

The 3rd stage, O-net is slightly different from that of P-Net and R-Net. 3 outputs been provided, which included: bounding box coordinates, facial landmarks and confidence level. Once the box had reach low confidence level and all the box been standardised, there should only be 1 bounding box for each faces at the end.

2.3.2 FaceNet

FaceNet is a unified embedding for face recognition approach, it directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity (Schroff et al., 2015). A vector of 128 numbers been output by FaceNet after user input the face image, the vector represents the most important features of a face. The vectors will then interpret as points in the Cartesian coordinate system. By doing that, the system could easily classify and recognise a people's face by calculating the distance of the vectors.

3 Description of work

This paper presented the implementation of face recognition using MTCNN and FaceNet into a real-life example, which is door lock mechanism. Besides, this approach also includes attendance tracking using google sheet as extra features. The project workflow was illustrated as shown in Figure 8.

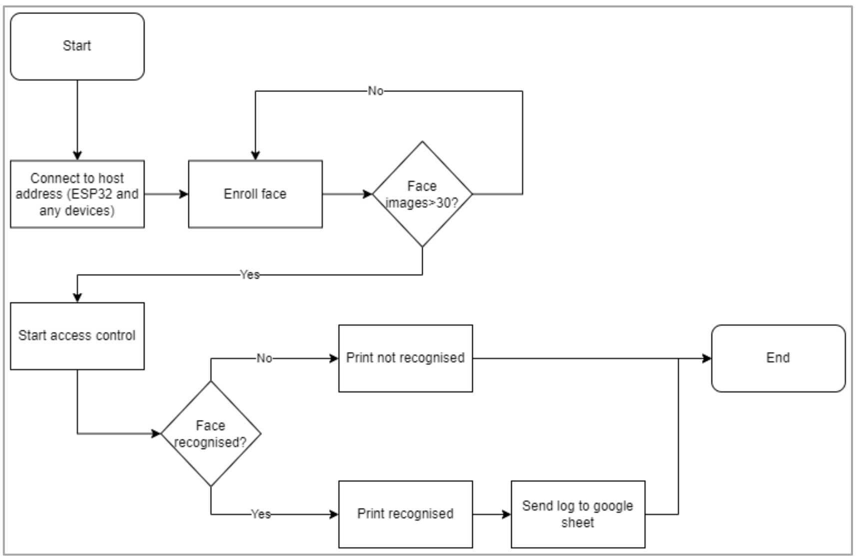


Figure 8: Work flowchart

4 Implementation

Chapter 4

4.1 Wiring connection

The wiring diagram below (Figure 9) shows the wiring connection for the approach (face recognition door lock). The figure was drawn using Fritzing and Inkscape. Fritzing is an open-source software tool used for electronic circuits design. It provides an intuitive interface for creating schematics, circuit diagrams, and PCB layouts, and allows users to easily share their designs with others.

While Inkscape helps to generate ESP32 cam modules with custom pin in .svg file. .svg file which are vector graphics format can be scaled up or down without losing quality or resolution. This allows custom module parts (ESP32 cam in this project) be easily resizable to accommodate different circuit layouts and sizes.

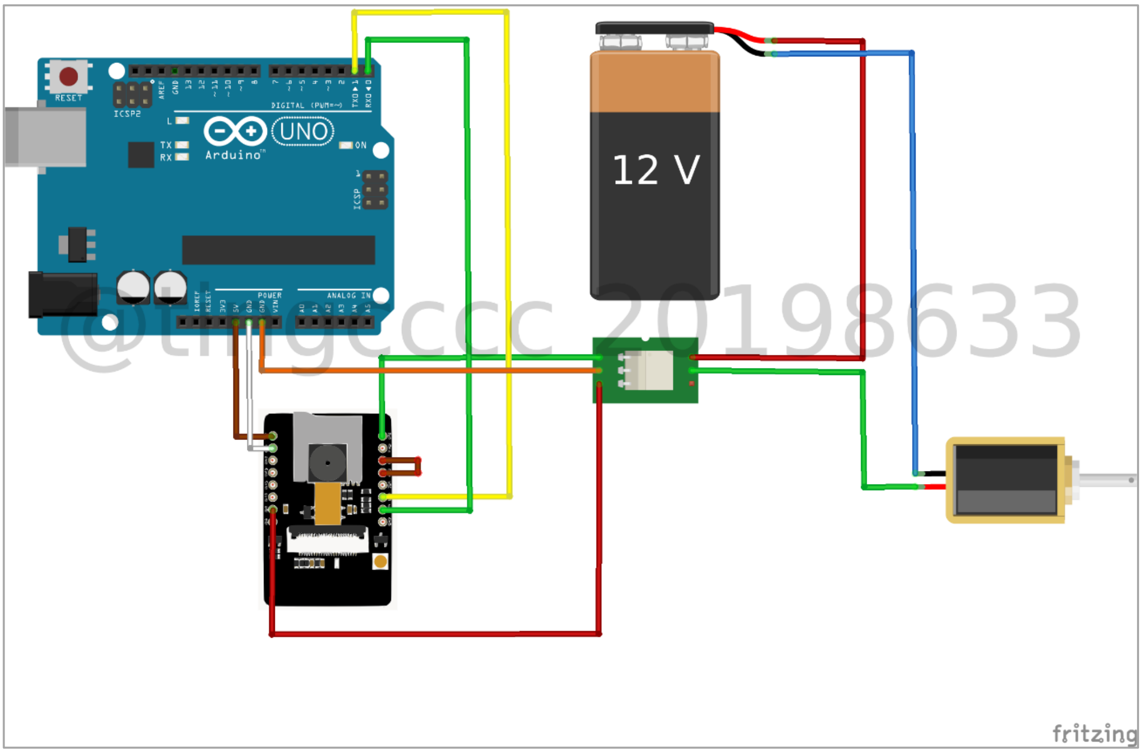


Figure 9: Wiring diagram of FRDL (face recognition door lock)

4.2 Partitions and libraries

As mentioned in Chapter 1 and Chapter 2.1, this project is implemented under Arduino IDE, which provides several pre-built partitions scheme and libraries inside. This approach required 2357027 bytes while doing code sketch, hence a maximum 3145750 bytes of partitions been created and named as “cc frdl scheme (3145750 bytes)” under path: “C:\Users\cc_fyp\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\tools\partitions”. Further details are included in the readme.txt file attached in the submitted zip file. The partition file is to specifies how the available flash and memory space on the microcontroller is divided between program code, data, and other essential components. In the other words, the partition in .csv format could allocate exact size of memory for microcontroller.

This project also uses arduinoWebSockets library to allow connection between ESP32 and devices been build using wireless network. This allows user to remotely access the view of ESP32 cam capture in real-time under same host address. Figure 10 shows the log shown when the connection established.

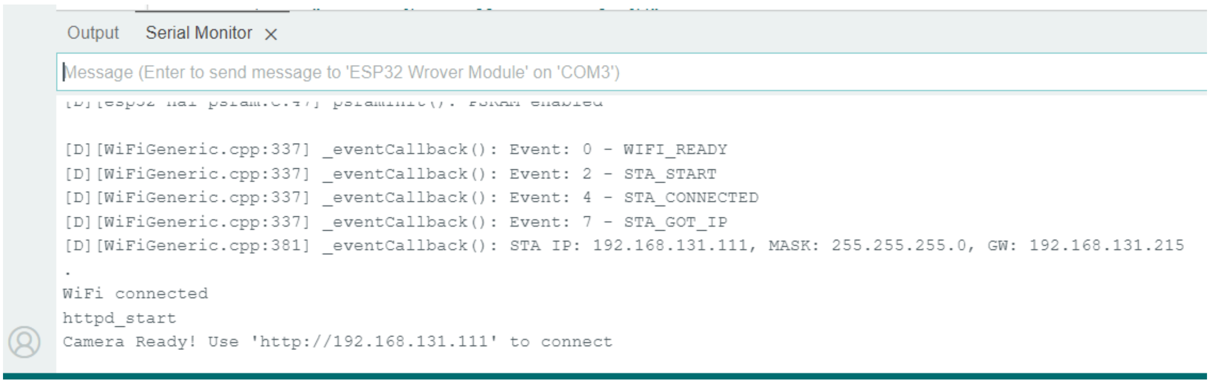


Figure 10: ESP32 connected, host address provided

4.3 Face recognition and access control

The architecture of face recognition applied had been discussed in Chapter 2.3. Thus, this Chapter 4.3 will provide a brief overview of the interface and its functionality. It is also good to note that the interface could simply change by adjusting the html file included and added in to replace the camera_index.h file. According to (WordBot, 2019), the adjusted html file needs to be converted into .h file which contains only HEX value before calling it in Arduino IDE .ino file. Appendix 1 shows the converted index file for interface.

Figure 11 shown below demonstrates the interface that appears after connecting to the same host address using any device.

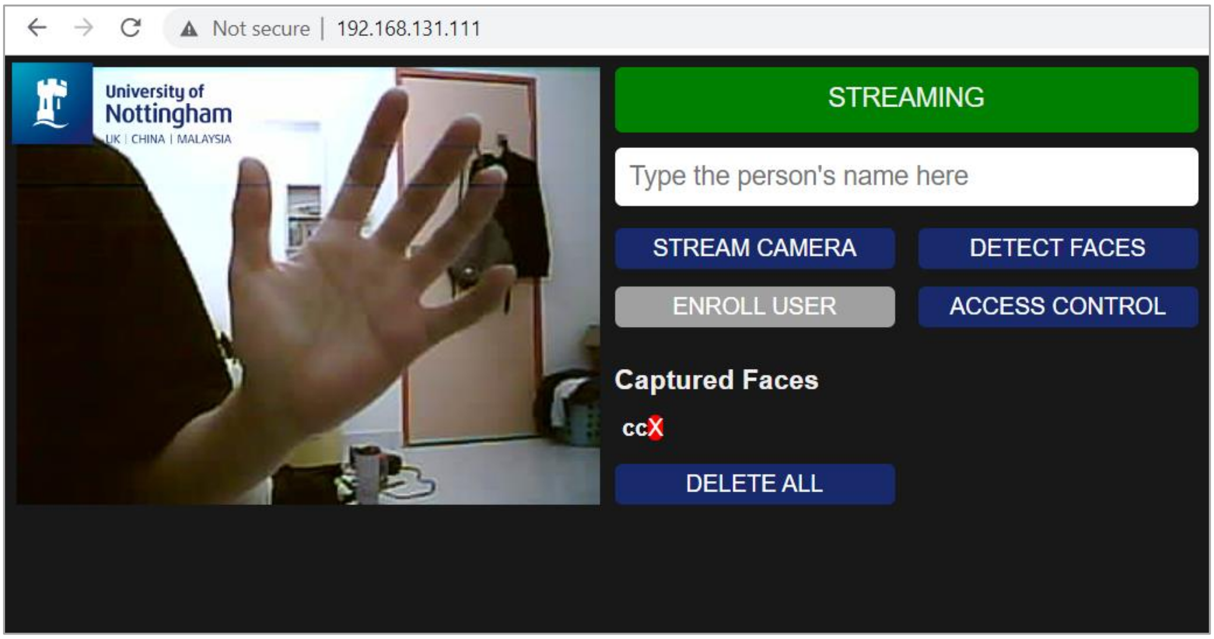


Figure 11: ESP32 cam Interface

In the leftmost window, the video captured in real-time by the ESP32 cam is displayed. There are four main functions included here, each represented by a different button. These functions include: streaming the camera, detecting faces, adding a user, and controlling access.

The "detect faces" button checks if any faces are detected, while the "add user" button only becomes active after a name has been entered into the corresponding field. Finally, the "access control" function starts the facial recognition process. The idea of the access control is relatively simple, it will send a string to webserver to notify user (Figure 12) and send a digital signal to relay once the face is recognised. The digital signal will energise the relay to connect both circuit and let the voltage pass through, which will unlock the solenoid door lock.

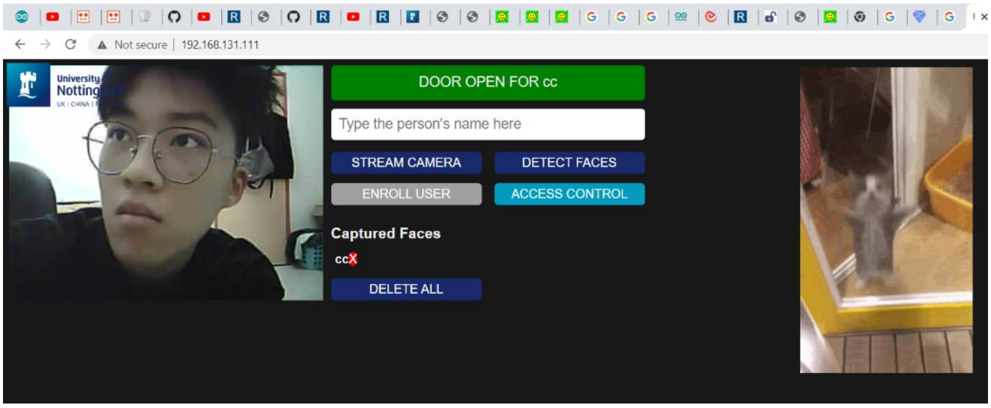


Figure 12: Door open for cc (enrolled username)

4.4 Attendance tracking

The attendance will be track by writing and sending the log to google spreadsheet online. Google app script, a scripting platform developed by Google for light-weight application development in the Google Workspace platform will be used to get the local date and time while the name will be get from Arduino IDE's parameter called "enroll_name", the parameter will be pass into function and then pass to app script before writing into google sheet. Figure 13 shows the attendance log data after face been recognised. Appendix 2 shows the code using in app script.

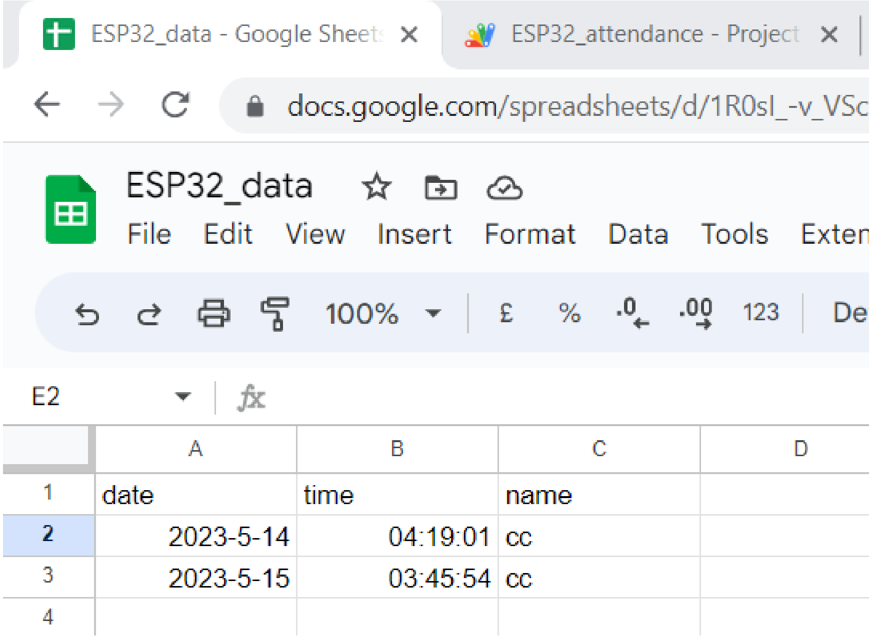


Figure 13: log data of attendee

5 Summary

In Chapter 5, the implementation and approaches used in this project will be summarized. Section 5.1 will provide a reflection on the implementation process, while section 5.2 will provide advice for those who wish to pursue further work on this project.

5.1 Reflection

This project offers an opportunity to delve deeper into the realm of intelligent agents, in this approach the intelligent part is the "face recognition" part which combine with real world applications and not just coding and software stuff. (Just kidding, coding and software could be fun :))

As conclusion, this approach still has a lot of potential and way to improve. For example, a more dependable model that incorporates a 3D face model could be applied.

5.2 Future works

There are several potential areas for future work in this project. First, the user interface could be enhancing to provide a more user-friendly system. Such as provides some instructions and guidance when user first entering the website. Besides, model evaluation should be conducted by doing evaluation and comparison on different models. Metrics such as loss and mean average precision should be consider and evaluate, which could help improving this face recognition approach. Lastly, the door lock mechanism should be integrated with the attendance tracking system. For instance, it should compare the current time with the scheduled time to determine whether to unlock the door or not, instead of simply unlocking the door when a recognized face is detected.

References

Gupta, P. (2023, February 27). *Types of arduino: Complete guide to 6 different types of Arduino*. EDUCBA. https://www.educba.com/types-of-arduino/

Watelectronics. (2021, October 14). *Arduino Uno Board : Features and real time applications*. WatElectronics.com. https://www.watelectronics.com/arduino-uno-board-tutorial-and-its-applications/

Team, T. A. (2023, April 26). *Arduino Uno Rev3 with long pins*. Arduino Documentation. https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins

Technology, A.-T. (2018). *ESP32-cam camera development board*. ESP32-CAM camera development board | 安信可科技. https://docs.ai-thinker.com/en/esp32-cam

justAuser, MELCHOR, N. R., Santos, S., Dennis, Esayas, M., Santos, R., Davies-Day, D., Alvim, E., Ignat, M., Budiwan, Ashish, Eduardo, Avi, Reynolds, N., Brabec, S., Shouman, W., Joey, ens4dz, H., B., … Shmuylovich, L. (2020, August 4). *ESP32-cam ai-thinker Pinout Guide: Gpios usage explained*. Random Nerd Tutorials. https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/

Identification, E. (2022). *Electronic identification*. Electronic IDentification - IDentity Verification Solutions. https://www.electronicid.eu/en/blog/post/face-recognition/en

Felix, K. (2023, April 22). *5 different types of facial recognition technology*. Techquirer. https://techquirer.com/5-different-types-of-facial-recognition-technology-and-their-evolution/#:~:text=5%20Different%20Types%20of%20Facial%20Recognition%20Technology%20and%20Their%20Evolution

Patil1, S. A. (2013). Face recognition : A S urvey - airccse.org. https://airccse.org/journal/ieij/papers/1113ieij05.pdf

Dulčić, L. (2020). *Face recognition with FaceNet and MTCNN*. – Ars Futura. https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/

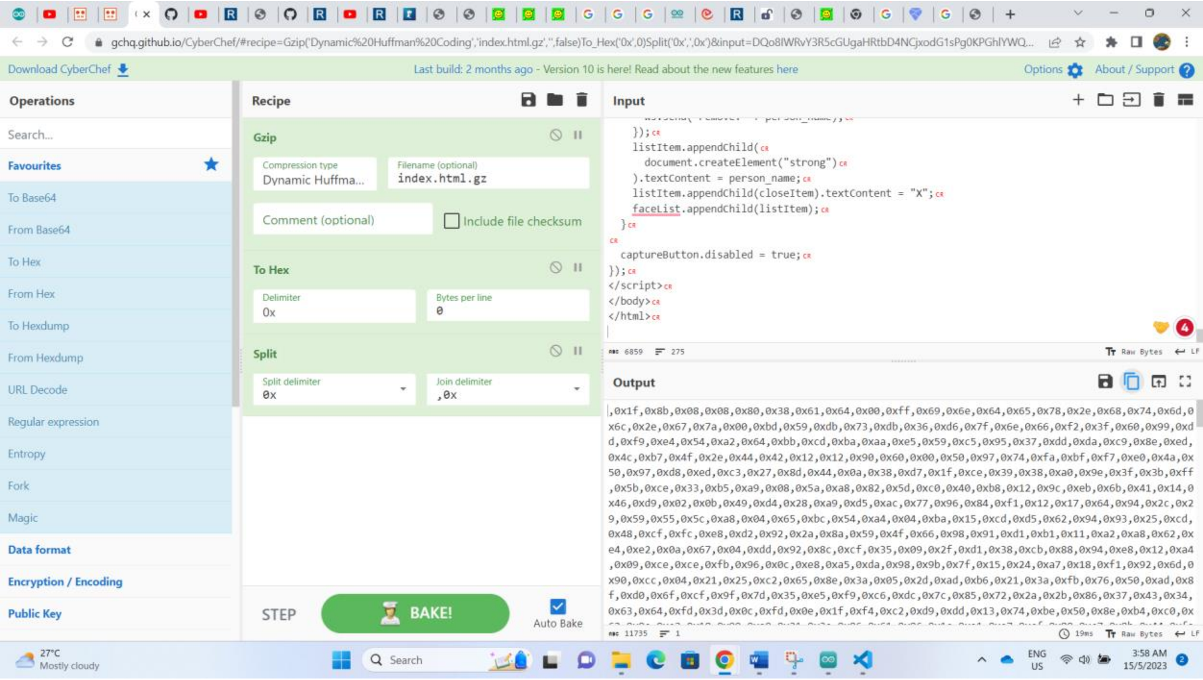
Wang, C.-F. (2018, July 27). *How does a face detection program work? (using neural networks)*. Medium. https://towardsdatascience.com/how-does-a-face-detection-program-work-using-neural-networks-17896df8e6ff

WordBot, A. (2019, April 25). *Editing the ESP32-cam camera web server HTML*. Robot Zero One. https://robotzero.one/esp32-cam-custom-html/

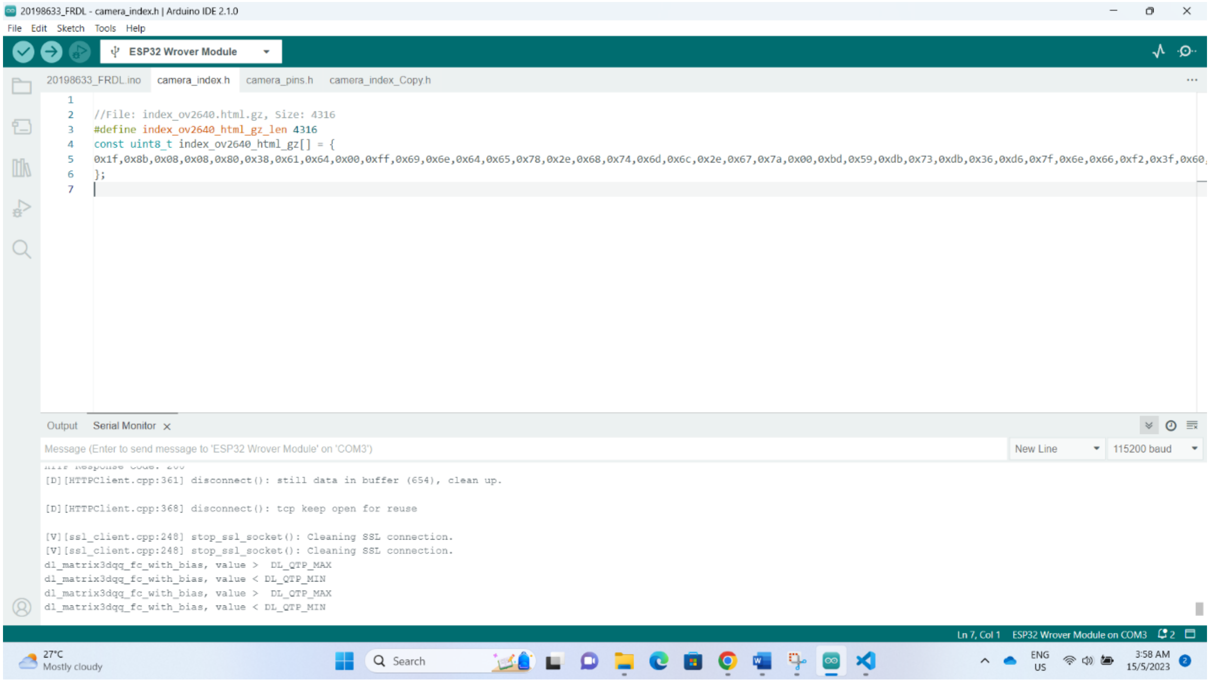
Schroff, F., Kalenichenko, D., & Philbin, J. (n.d.). *FaceNet: A Unified embedding for face recognition and clustering*. FaceNet: A Unified Embedding for Face Recognition and Clustering. https://arxiv.org/pdf/1503.03832.pdf

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503. https://doi.org/10.1109/lsp.2016.2603342

Appendix 1

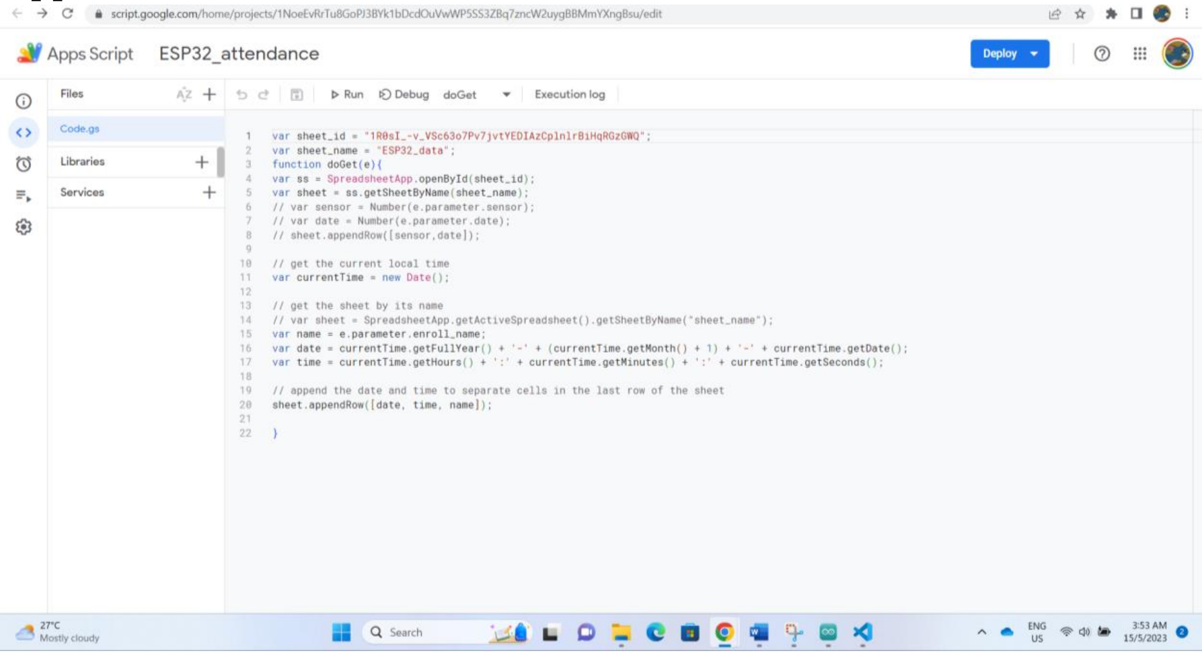


(Convert into HEX value)



(Add into the same folder of .ino file)

Appendix 2



(Google App Script that collect local date time and passing parameters to sheets)