

CSC1250 Capstone Project

DIT 2205

Team Members



Lee Wei Yi



Tan Ming Reo



Problem Statement

The current car park system at MCKL college lacks efficiency as it relies on manual authentication by guards. This process often leads to delays and uncertainties, as guards cannot always ensure that the vehicle entering the premises belongs to a student or authorized personnel.

Automated Smart Entry System



Our project focuses on the development of an Automated Smart Entry System. This system combines **facial recognition** and **car plate recognition** technologies with **seamless hardware integration**.



Aim

To develop an integrated parking system that combines facial recognition and license plate recognition technologies to enhance security, improve entry processes, and enable data-driven management of parking infrastructure

Objectives



Develop and integrate facial recognition technology to accurately identify authorized individuals for parking access.



Implement license plate recognition technology to automate entry processes and enhance security by verifying vehicle registrations.



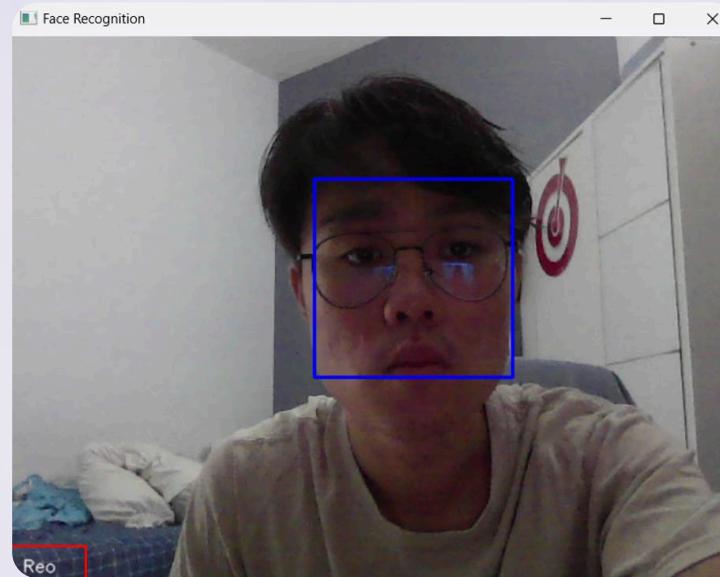
Design and deploy a user-friendly hardware and software system for real-time processing of facial and license plate data, ensuring efficient and reliable recognition.



Features

1

Facial Recognition



Advanced facial recognition cameras identify authorized individuals, ensuring secure access control.

2

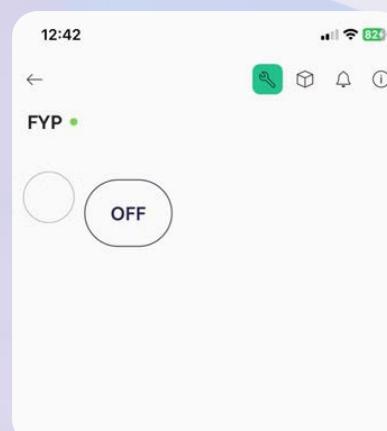
License Plate Scanning



License plate readers automatically verify registered vehicles, streamlining the entry process.

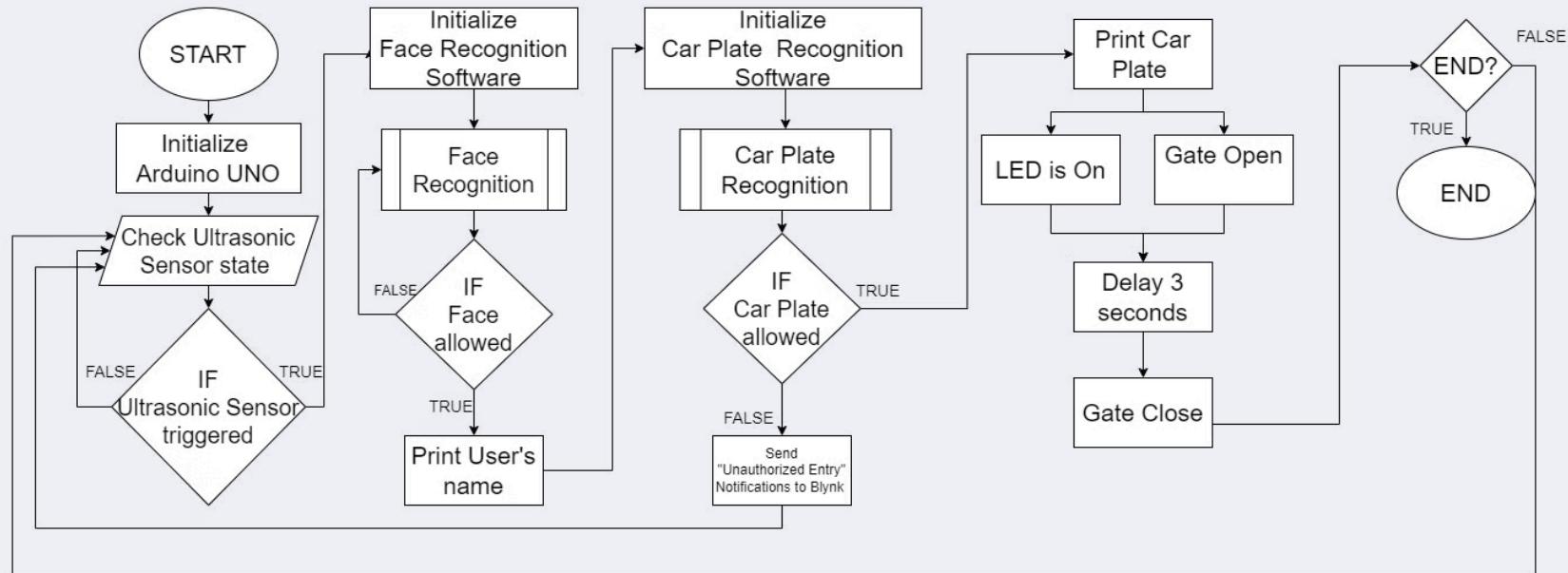
3

IoT Connectivity

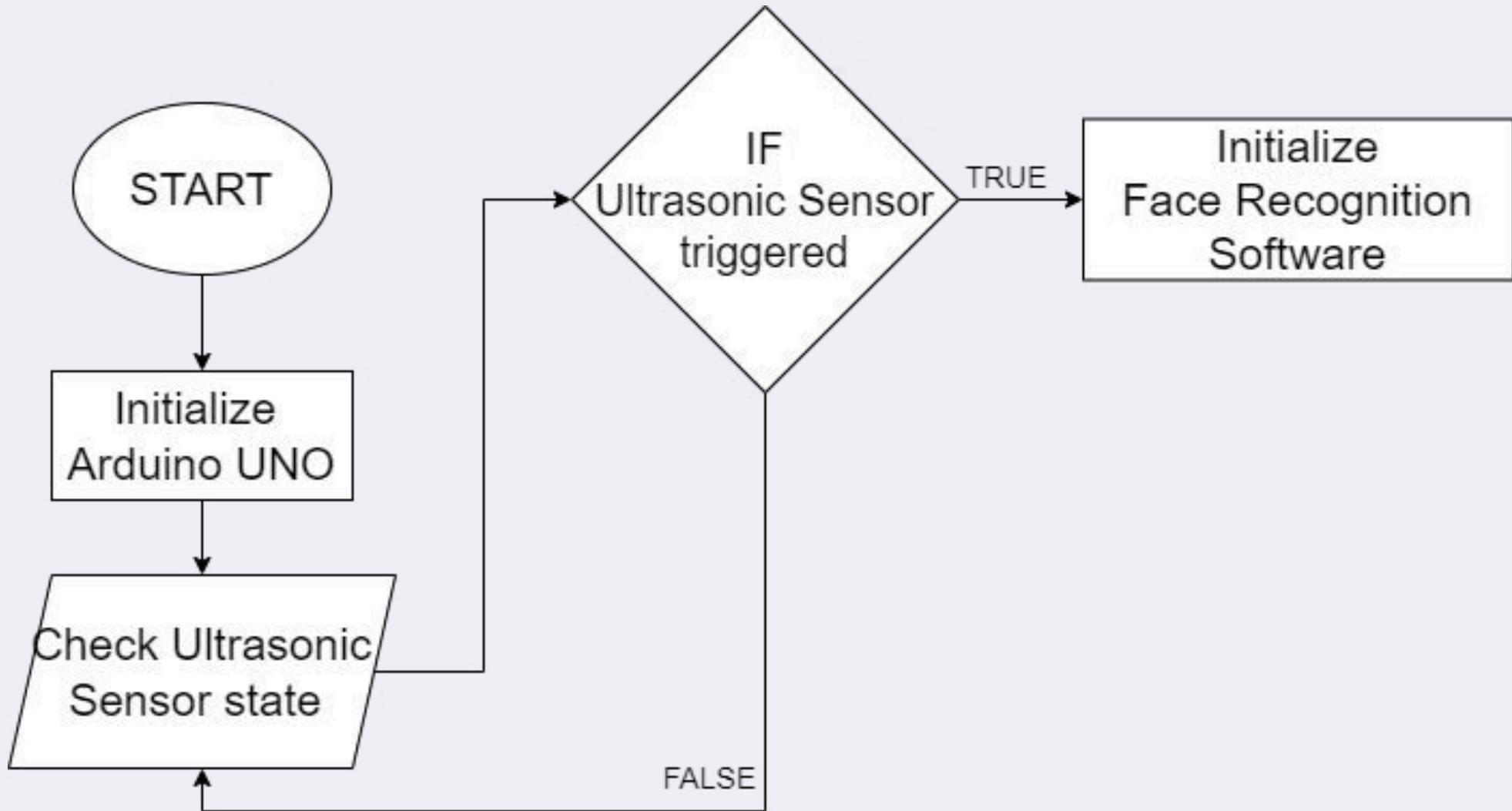


The system integrates with the Blynk IoT platform for remote monitoring and control capabilities.

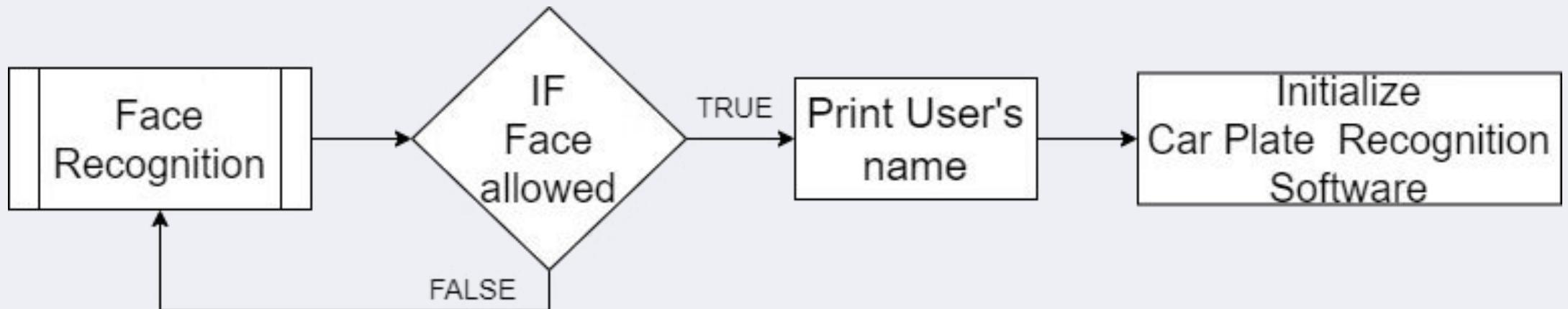
Flow Chart



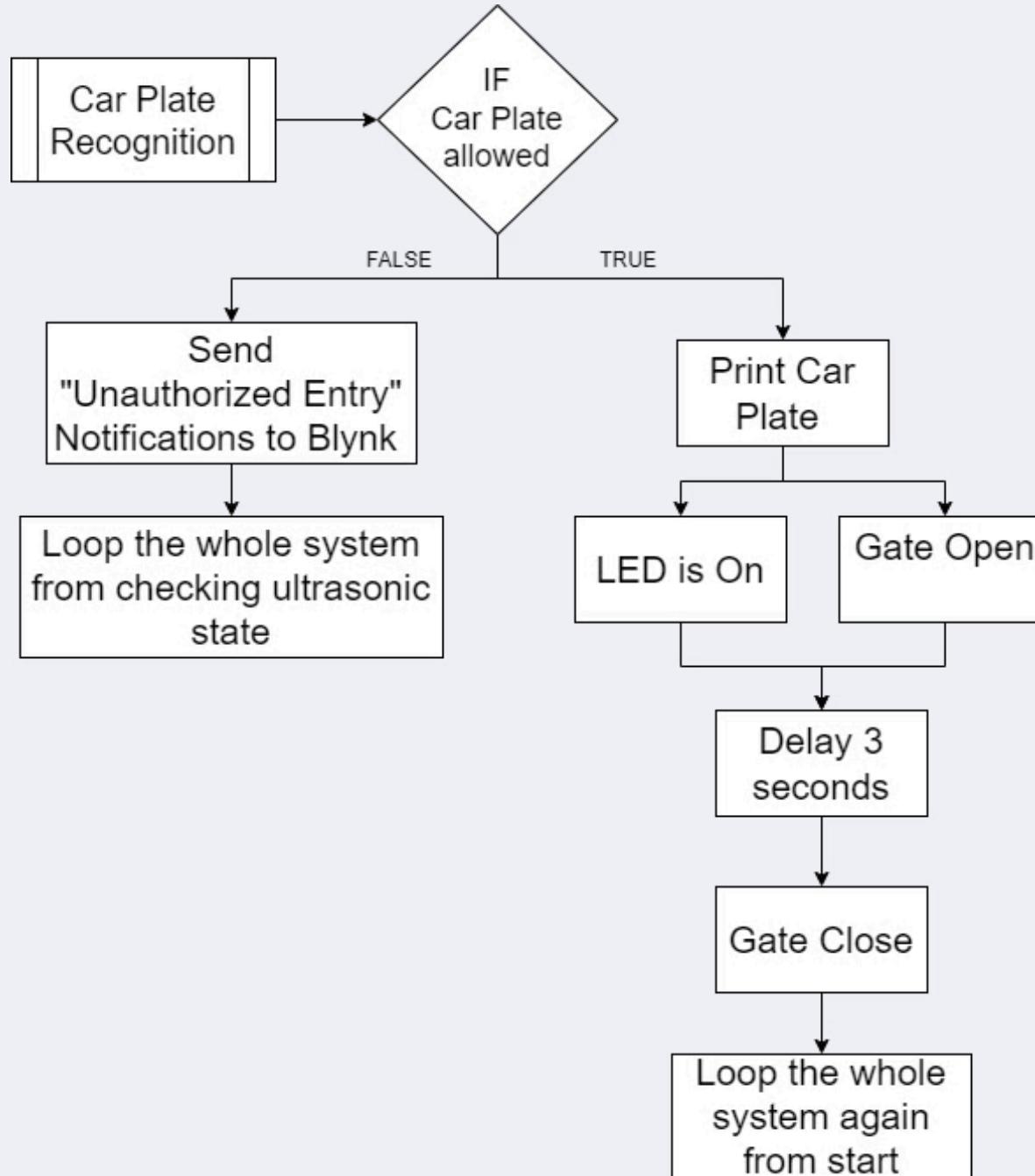
Flow Chart



Flow Chart



Flow Chart



Platform

PyCharm

A screenshot of the PyCharm IDE interface. The left sidebar shows the project structure with files like main.py, capture.py, Car.py, Livefeed.py, and requirements.txt. The main code editor window displays Python code for a car recognition application using YOLOv3 and OpenCV. The code includes imports for os, cv2, numpy, matplotlib.pyplot, util, easyocr, and openpyxl. It defines constants for model paths, allowed plates, and loads class names from a file. The code then reads images from a directory, loads the model, and performs object detection.

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import util
import easyocr
from openpyxl import Workbook

# define constants
model_cfg_path = os.path.join('..', 'model', 'cfg', 'darknet-yolov3.cfg')
model_weights_path = os.path.join('..', 'model', 'weights', 'model.weights')
class_names_path = os.path.join('..', 'model', 'class.names')

input_dir = "C:\\Users\\Wei Yi\\PycharmProjects\\CarRecognition2\\yolov3-from-opencv-object-detection-master\\Car Image"

# Assuming you have a database of allowed car plates stored in a dictionary
allowed_plates = {
    "PJP 1382": "Alex Lee",
    "RX 6326": "Shermaine Tan",
    # Add more entries as needed
}

for img_name in os.listdir(input_dir):
    img_path = os.path.join(input_dir, img_name)

    # load class names
    with open(class_names_path, 'r') as f:
        class_names = [j[:-1] for j in f.readlines() if len(j) > 2]
    f.close()

    # load model
    net = cv2.dnn.readNetFromDarknet(model_cfg_path, model_weights_path)
```

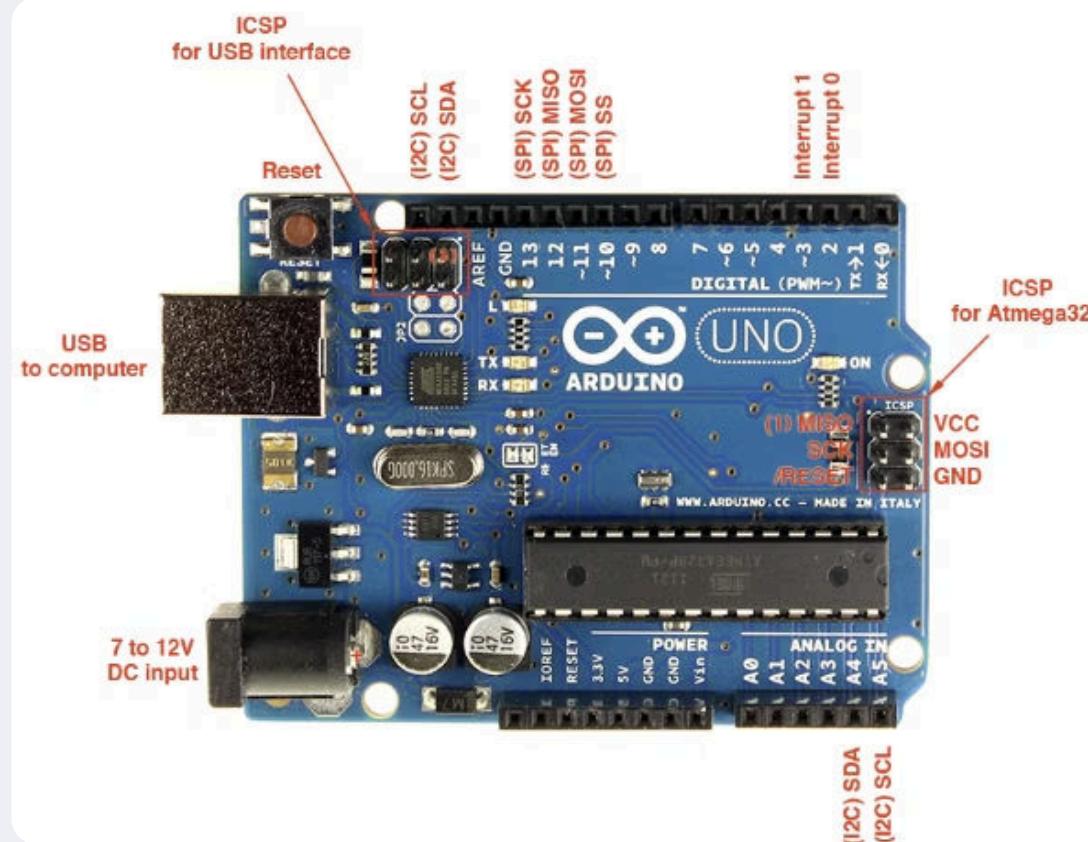
Arduino IDE

A screenshot of the Arduino IDE. The top menu bar shows 'File Edit Sketch Tools Help'. The main workspace shows a sketch named 'sketch_may07a'. The code contains two functions: setup() and loop(). The setup() function has a comment // put your setup code here, to run once;. The loop() function also has a comment // put your main code here, to run repeatedly;. The code is written in C++/Arduino syntax.

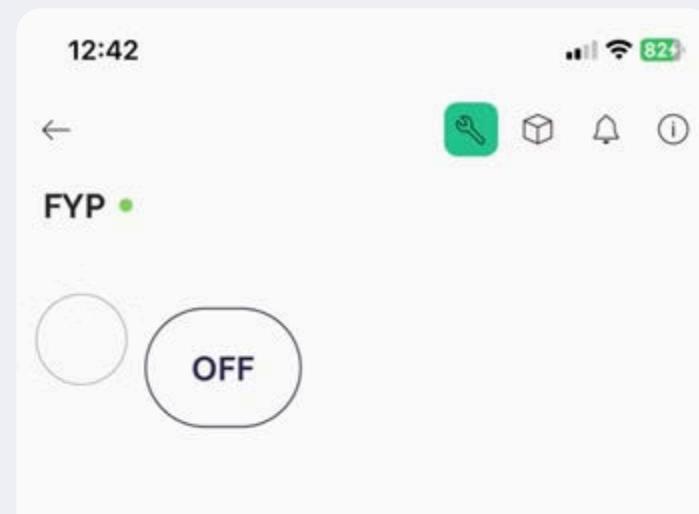
```
sketch_may07a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_may07a
void setup() {
  // put your setup code here, to run once:
}

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

Arduino Uno



Blynk



My Roles

Car Plate Recognition

- **Designed and implemented** the car plate recognition system.
- **Ensured system efficiency** for reliable identification and processing of license plate information.

Arduino Hardware

- **Set up** essential Arduino hardware components.
- Included ultrasonic sensors, servo motors, LEDs, and buttons.
- **Ensured smooth interaction between hardware and software** for automated entry processes.

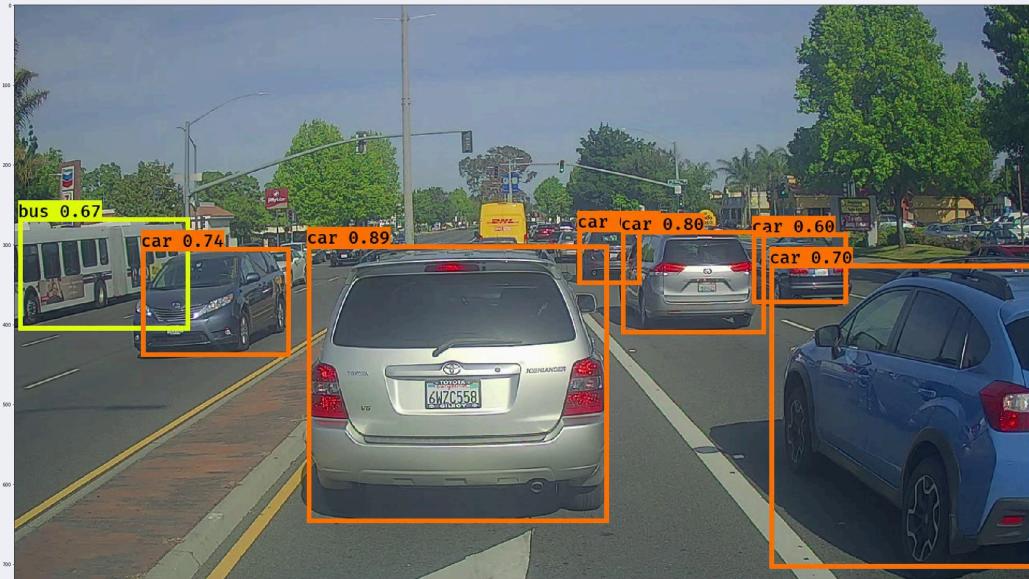


Car Plate Recognition

The Automated Smart Entry System leverages advanced license plate recognition technology to automatically verify registered vehicles.



Technologies



YOLO (You Only Look Once)

YOLO is an **object detection algorithm** that allows us to detect and localize license plates within images.



EasyOCR

EasyOCR is a Python library that provides optical character recognition (OCR) capabilities, allowing us to extract text from license plate images.

Why do we choose YOLO?

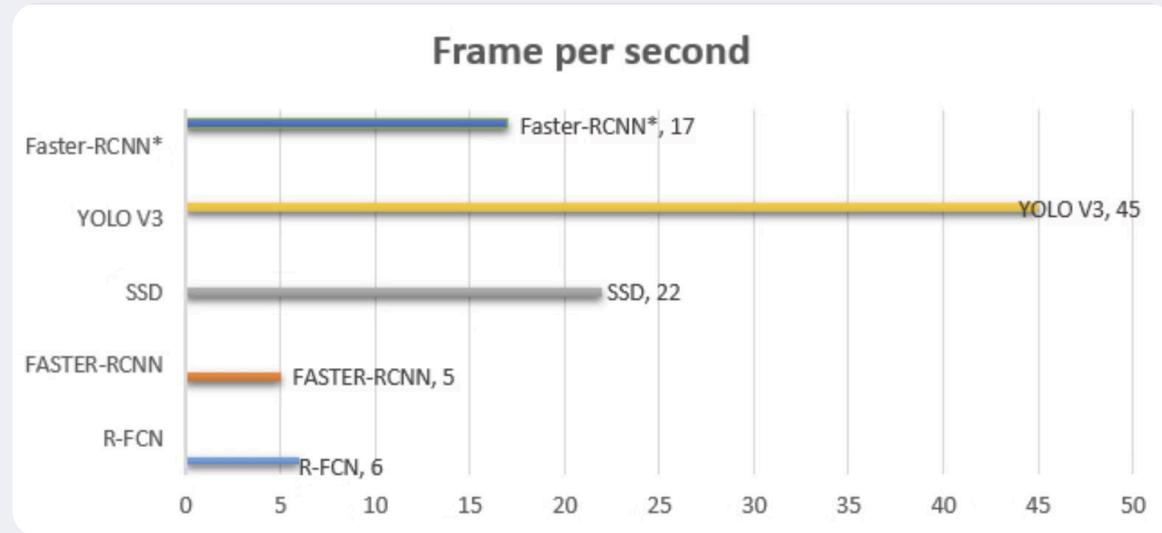


Figure 1.0: Comparison of object detection models in term of speed.

1. **Real-time Performance:** YOLO swiftly detects objects, crucial for our parking system's efficiency.
2. **Accuracy:** YOLO maintains accuracy while being fast, ensuring reliable license plate recognition.
3. **Open Source:** YOLO's open-source nature provides resources and support for ongoing improvement.

How Train Yolo Model?



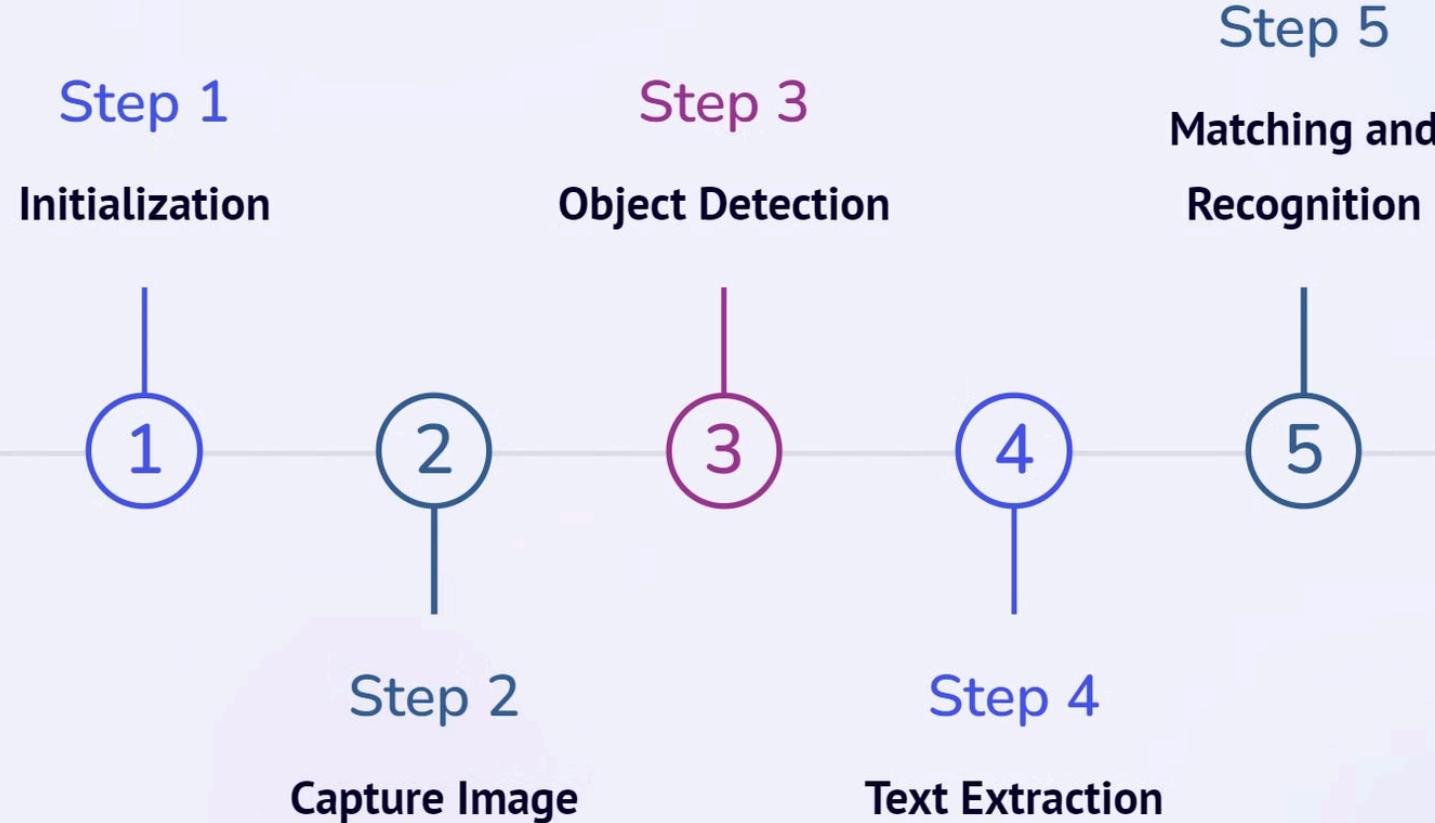
How Yolo detect Car Plate?



How Yolo detect Car Plate?



Flow of Car Plate Recognition System



Step-by-Step Guide

Step 1: Initialization

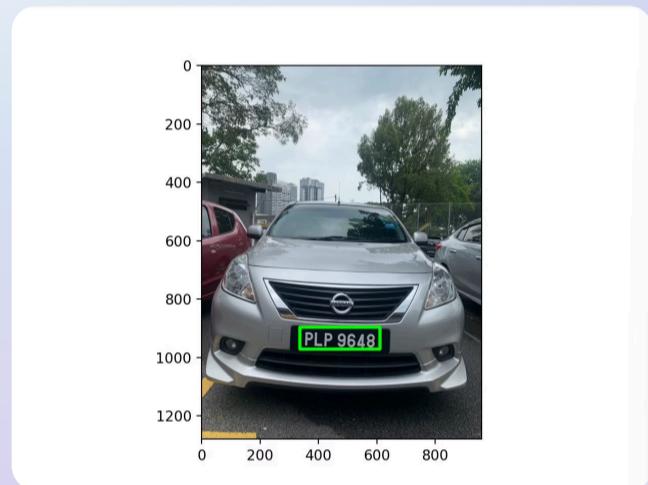
1

- Load YOLO model for object detection.
- Load EasyOCR reader for text extraction.

Step 3: Object Detection

3

- Use YOLO for car detection in the image.
- Extract bounding boxes and confidence scores for detected cars.



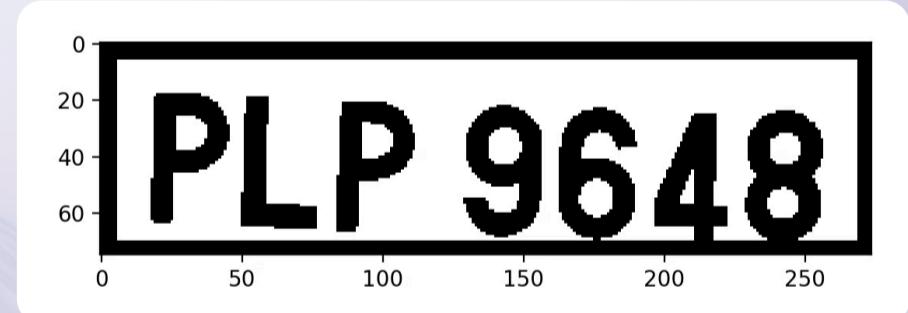
Step 5: Matching and Recognition

- Match the extracted text with the pre-defined list of allowed plates.
- If a match is found, label the plate as allowed.
- If no match is found, label the plate as unauthorized user and deny entry.

4

Step 4: Text Extraction

- Extract text from the detected car plates using EasyOCR.
- Process the extracted text and confidence scores.



5

Step 6: Output

- Display the recognition message and status on the user interface.
- Update the system status and log the information.

Not allowed:

PLP 9648 0.9099767950444997
Unauthorized Car: License Plate - PLP 9648

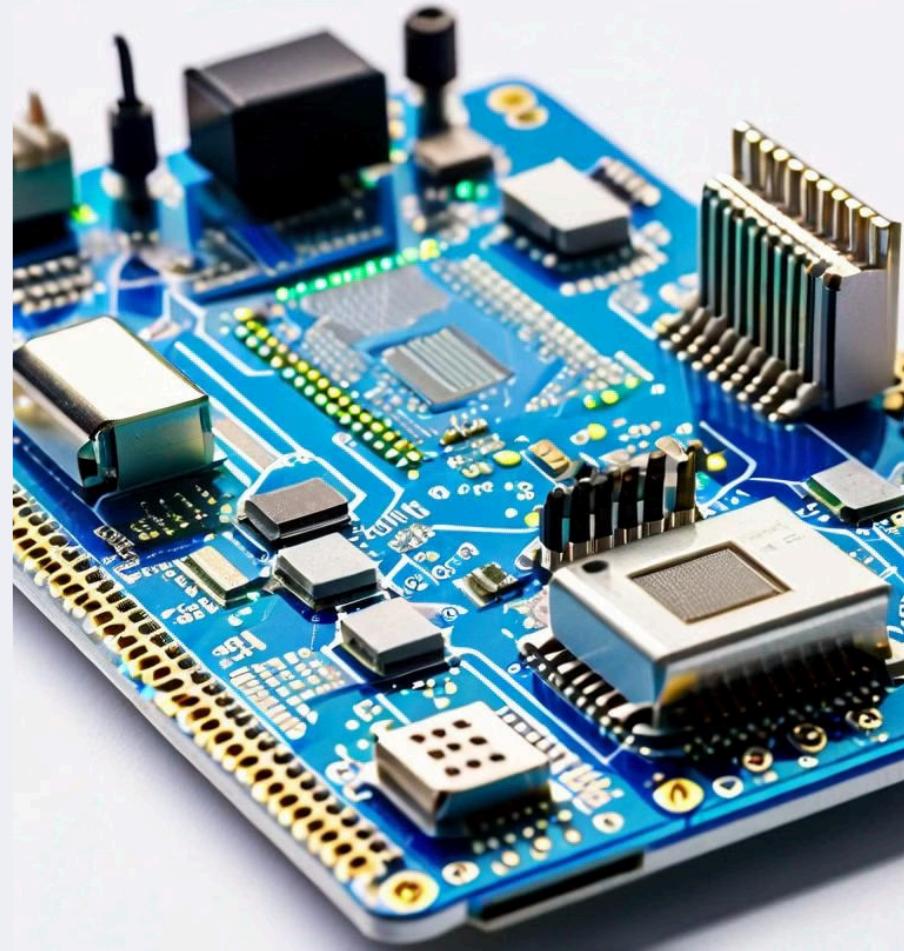
Allowed:

PJP 1382 0.9575462737961343
Allowed Entry

Arduino UNO

The Arduino Uno is a widely-used microcontroller board recognized for its simplicity and versatility, designed for students, educators, and professionals.

- **User-Friendly**
- **Open-Source Ecosystem**
- **IoT Applications**



Components



Ultrasonic Sensor

Detects the presence of cars, in front of the barrier by emitting ultrasonic waves and measuring their reflection.



Servo Motor

Acts as the barrier mechanism, controlling the opening and closing of the entry point.



Button

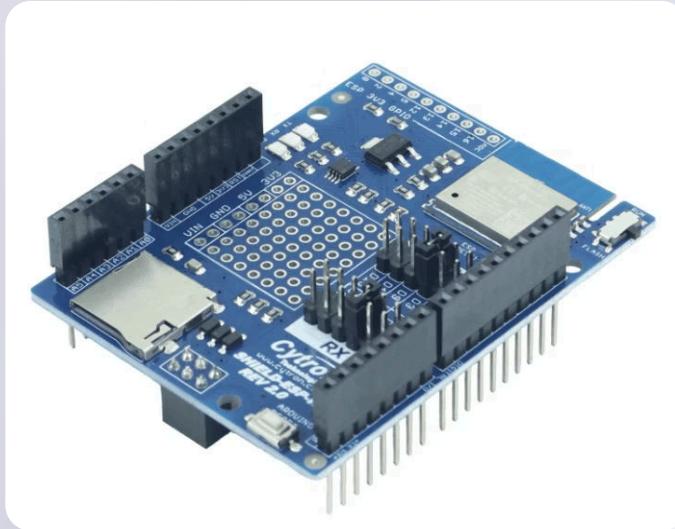
Offers manual override capability, allowing users to manually control the barrier when necessary.



LED

Provides visual indication of the system's status, particularly when an object is detected near the barrier.

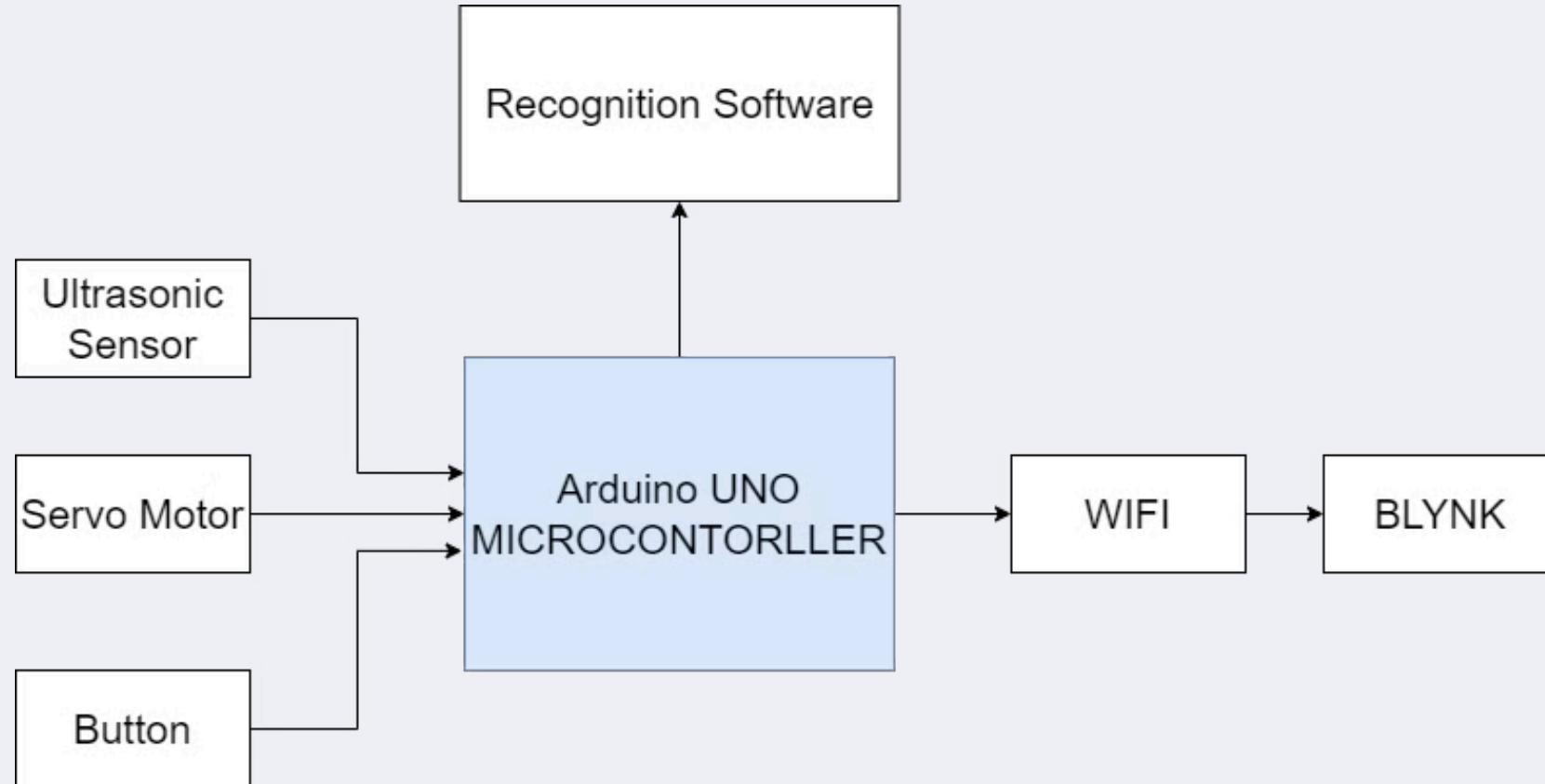
Components



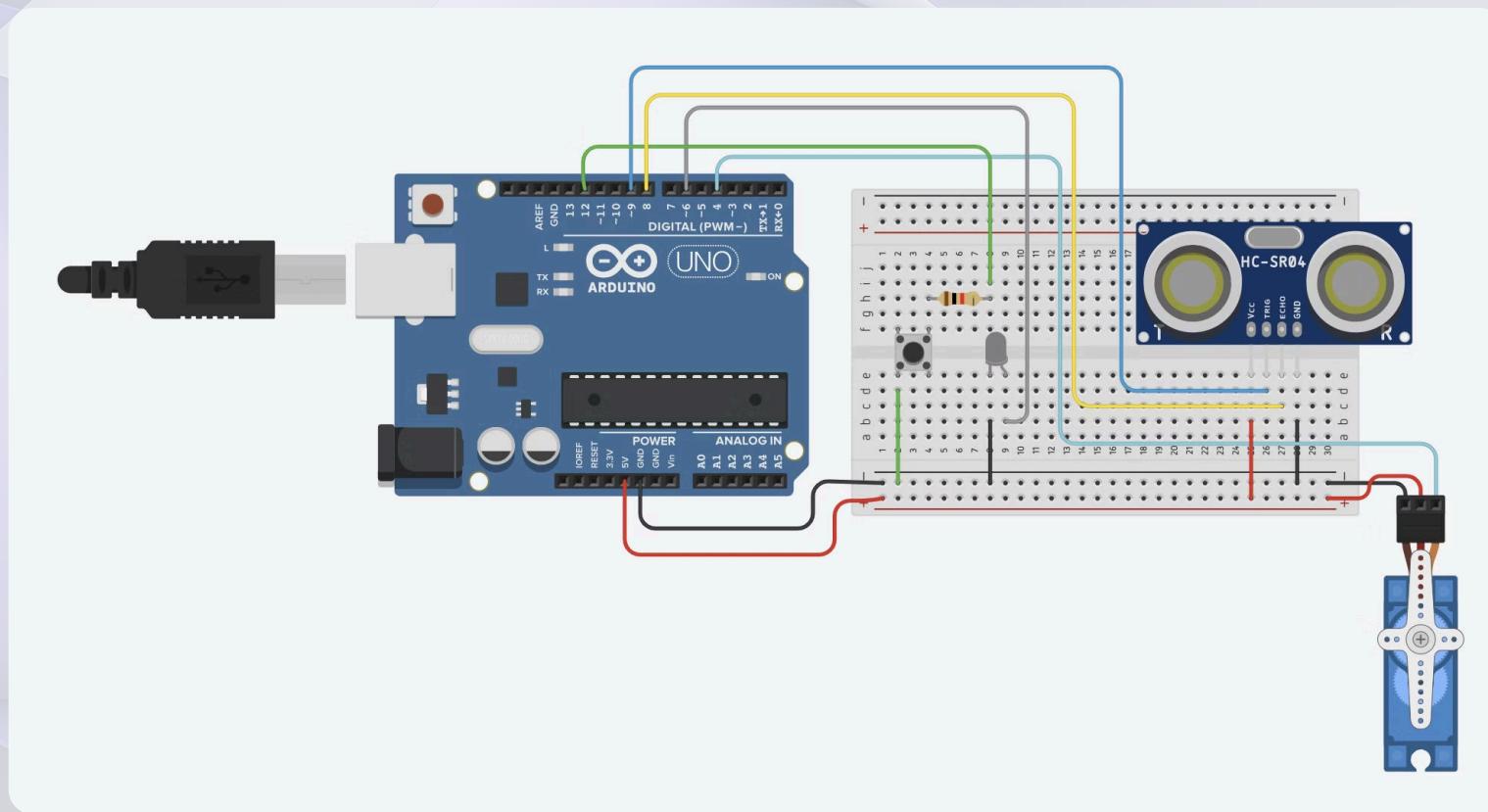
ESP8266 WiFi-Shield

The ESP8266 module acts as a Wi-Fi shield, **enabling wireless connection for the Arduino Uno**. It connects the system to the Blynk platform, allowing remote monitoring and control via the internet and the Blynk mobile app.

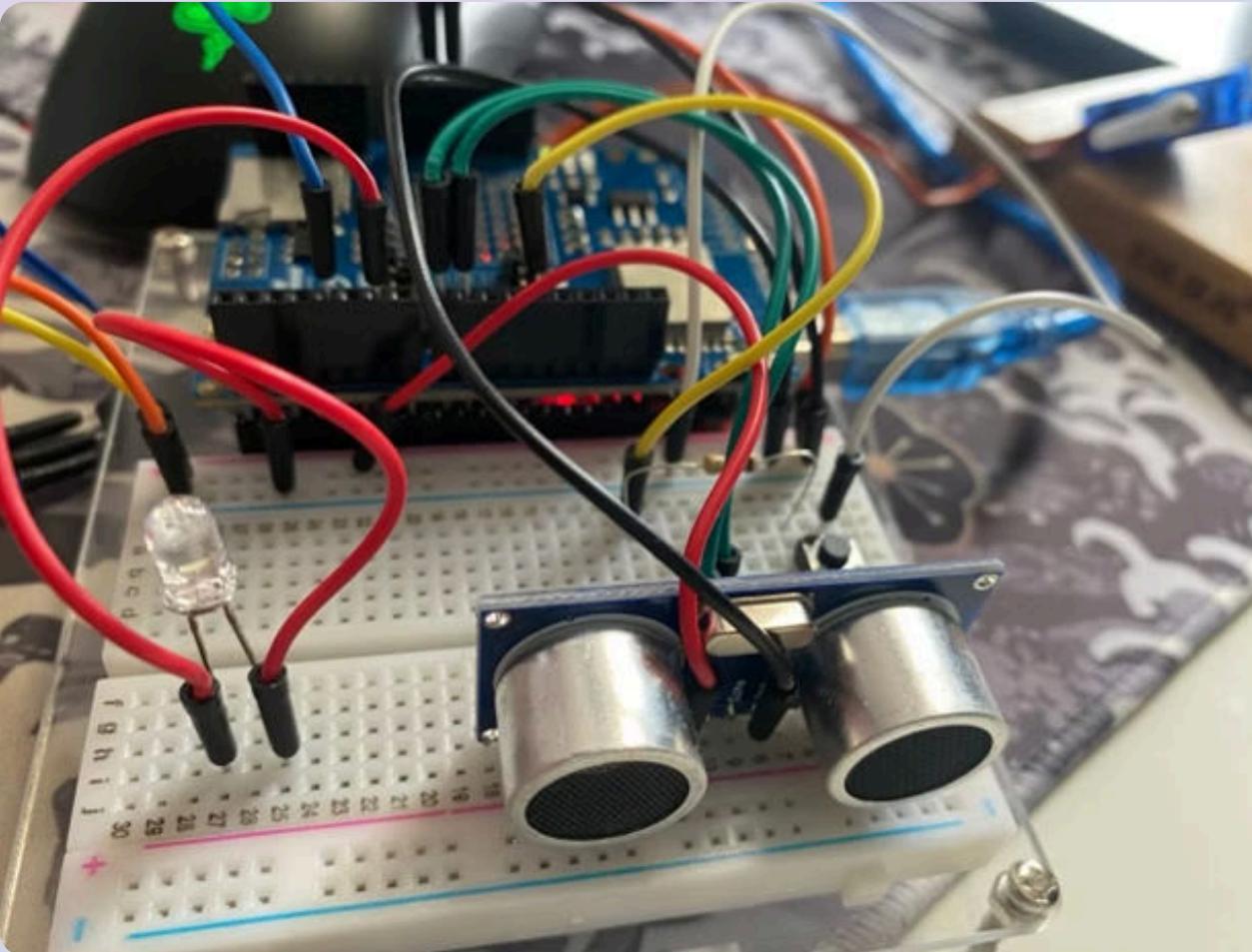
Block Diagram



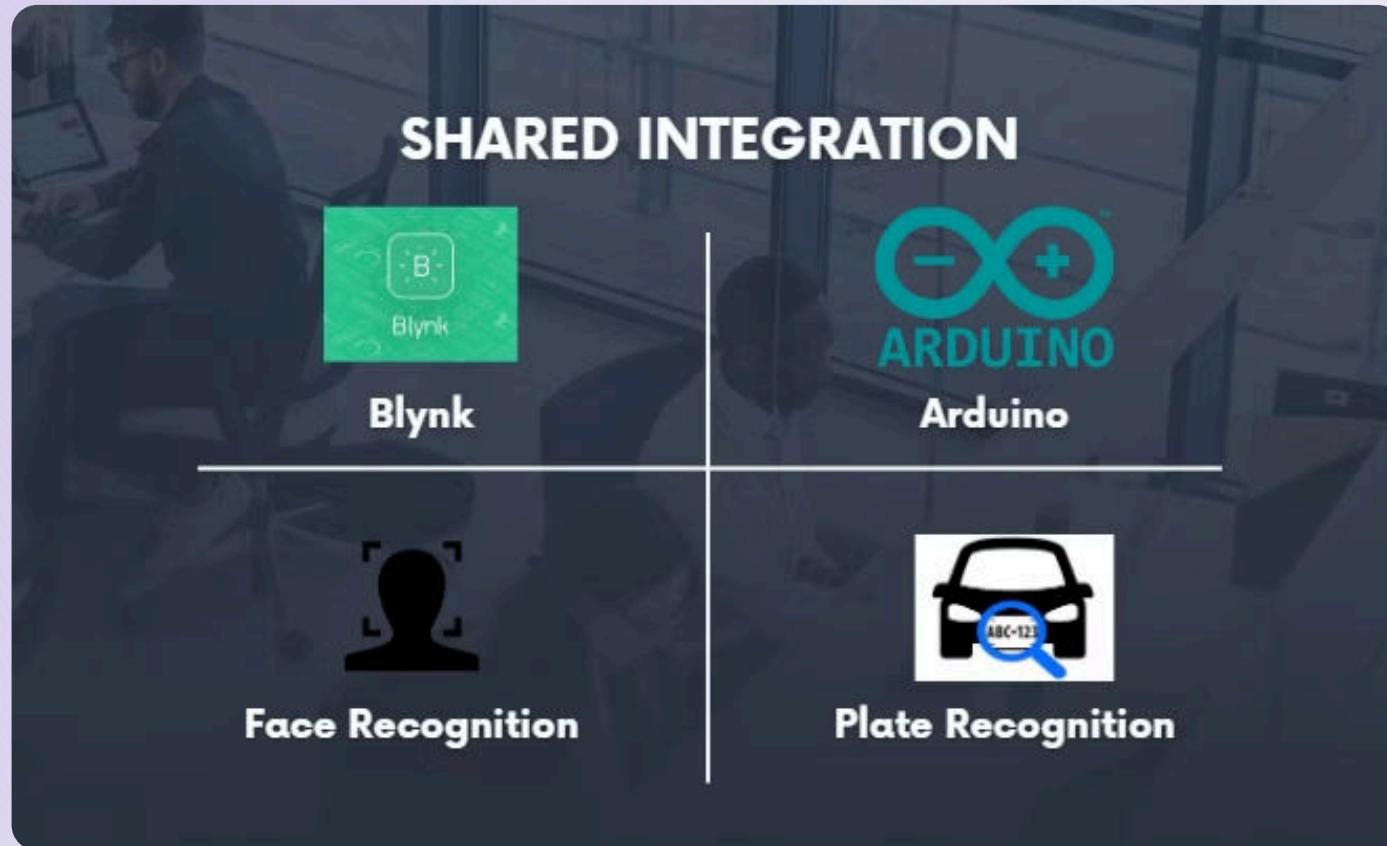
Wiring Diagram



Arduino Connections



Integration



Conclusion

By blending **facial and license plate recognition**, we **boost security and simplify processes**. We've used top-notch tools like **PyCharm, Arduino Uno, and Blynk platforms** to showcase our dedication. Our hardware solutions, including **ultrasonic sensors, servo motors, LEDs, and buttons**, demonstrate our commitment.

Overall, our project **shows how technology can improve access control**, highlighting our team's skills. We expect similar systems to become widespread, changing how we manage and secure spaces.

Q&A



Thank you