Access Control System for Gates with License Plate Recognition

Ventura Rafael, Crespo Francisco

ESTGOH, Oliveira do Hospital, Portugal

a2022131397@alnuos.estgoh.ipc.pt, a2021138307@alnuos.estgoh.ipc.pt

Links: Rafael Ventura, Francisco Crespo

Abstract. This study presents the design and implementation of an access control system for gates based on license plate recognition. Leveraging affordable technologies such as Arduino microcontrollers, ESP32-CAM, OpenCV, and Tesseract OCR, the system identifies vehicles by processing captured images. Key features include real-time vehicle detection using ultrasonic sensors, a potentiometer to adjust detection distance, a Flask API for image processing and data persistence, and validation of license plates based on Portuguese patterns. The proposed system effectively automates gate access, offering a scalable and cost-efficient solution for residential and commercial security.

Keywords: Access Control · License Plate Recognition · Arduino · OpenCV · Automation.

Introduction

Security in access control is a critical concern in residential and commercial contexts. Traditional methods relying on manual gate operation are often inconvenient and prone to human error. Automated systems using license plate recognition offer a modern alternative, improving convenience and security. The objective of this study is to develop a system that automates gate access using a combination of hardware components (Arduino, ESP32-CAM, potentiometers and Ultrasonic sensor) and software tools (OpenCV and Tesseract). The system recognizes Portuguese license plates and validates access based on an encrypted database of authorized plates. By addressing existing challenges in accuracy and cost-effectiveness, this work aims to contribute to the growing field of smart security systems.

Literature Review

Existing research highlights the use of OpenCV and Tesseract for text recognition in various domains. Automated license plate recognition (ALPR) systems have been explored but often require expensive hardware and advanced computational resources. For example, studies have shown that integrating ALPR systems with IoT frameworks can significantly enhance functionality. However, many of these implementations lack localized validation algorithms tailored to specific license plate patterns. This paper fills the gap by developing a low-cost ALPR system that validates Portuguese license plates using custom algorithms. The proposed solution integrates hardware and software to deliver real-time access control efficiently.

Methodology

System Architecture

The system is based on the following components:

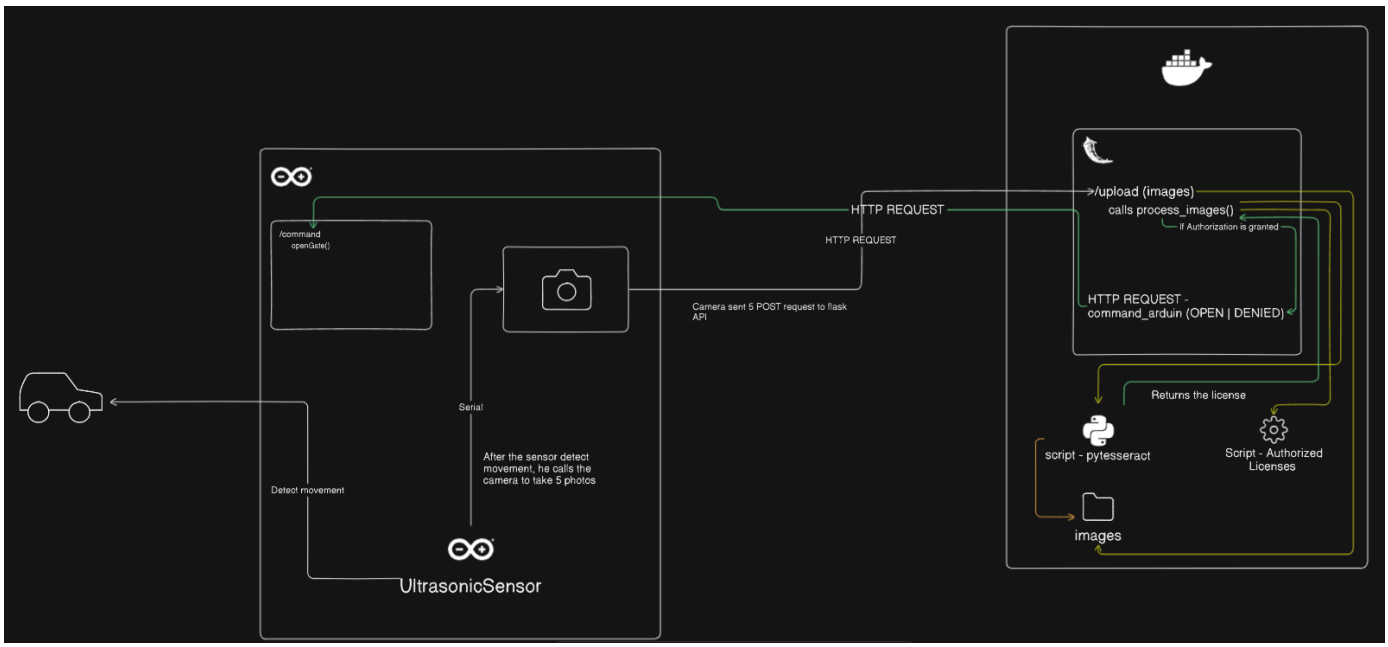

Figure 1: System Architecture Overview.

- **Arduino Rev2 WiFi:** Coordinates the workflow and handles communication.
- **ESP32-CAM:** Captures images upon motion detection. *(Santos Rui, 2019)*
- **Ultrasonic Sensor:** Detects the presence of vehicles.
- **Potentiometer:** Allows dynamic adjustment of the ultrasonic sensor's detection range.
- **Servo Motor:** Controls gate movement.
- **IO Extension Shield:** Enables multiple peripheral connections.

Integrated System Characteristics: The Arduino Rev2 WiFi acts as the central controller, orchestrating the overall workflow by managing communication between the ultrasonic sensor, ESP32-CAM, and servo motor. This centralized control ensures that the system operates as a cohesive unit, minimizing complexity in component interactions.

Distributed System Characteristics: The delegation of image processing to a Flask API hosted in a Docker container introduces a distributed element. While the ESP32-CAM captures the images locally, the computationally intensive task of image recognition is offloaded to a remote server. This separation of responsibilities enhances system efficiency and allows for scalability.

The balance between integration and distribution ensures the system is both cost-effective and adaptable to various applications. This combination of centralized coordination and distributed processing defines the system as a hybrid architecture, leveraging the advantages of both approaches to deliver efficient and scalable performance.

Communication Between Components

Detection: The ultrasonic sensor, connected to the Arduino, detects objects within a distance dynamically set via the potentiometer.

Image Capture: The Arduino sends a serial command to the ESP32-CAM to capture five images.

Image Processing: The ESP32-CAM sends the images via HTTP to a Flask API hosted in a Docker container.

License Plate Recognition: The API processes the images to extract license plate information using OpenCV and Tesseract. *(Ahmed Boktiar, 2023)*

Validation and Authorization

Extracted plates are validated against the Portuguese format (e.g., two pairs of numbers and one pair of letters). A custom algorithm ensures the most accurate plate is selected. If the plate is authorized, the system sends a command to the Arduino to open the gate. If not, the action is denied.

Safety Mechanism

Before closing the gate, the ultrasonic sensor checks for obstacles, ensuring the system avoids accidental damage.

Electrical Circuit Design

This subsection outlines the electrical connections between the system components. The circuit design includes:

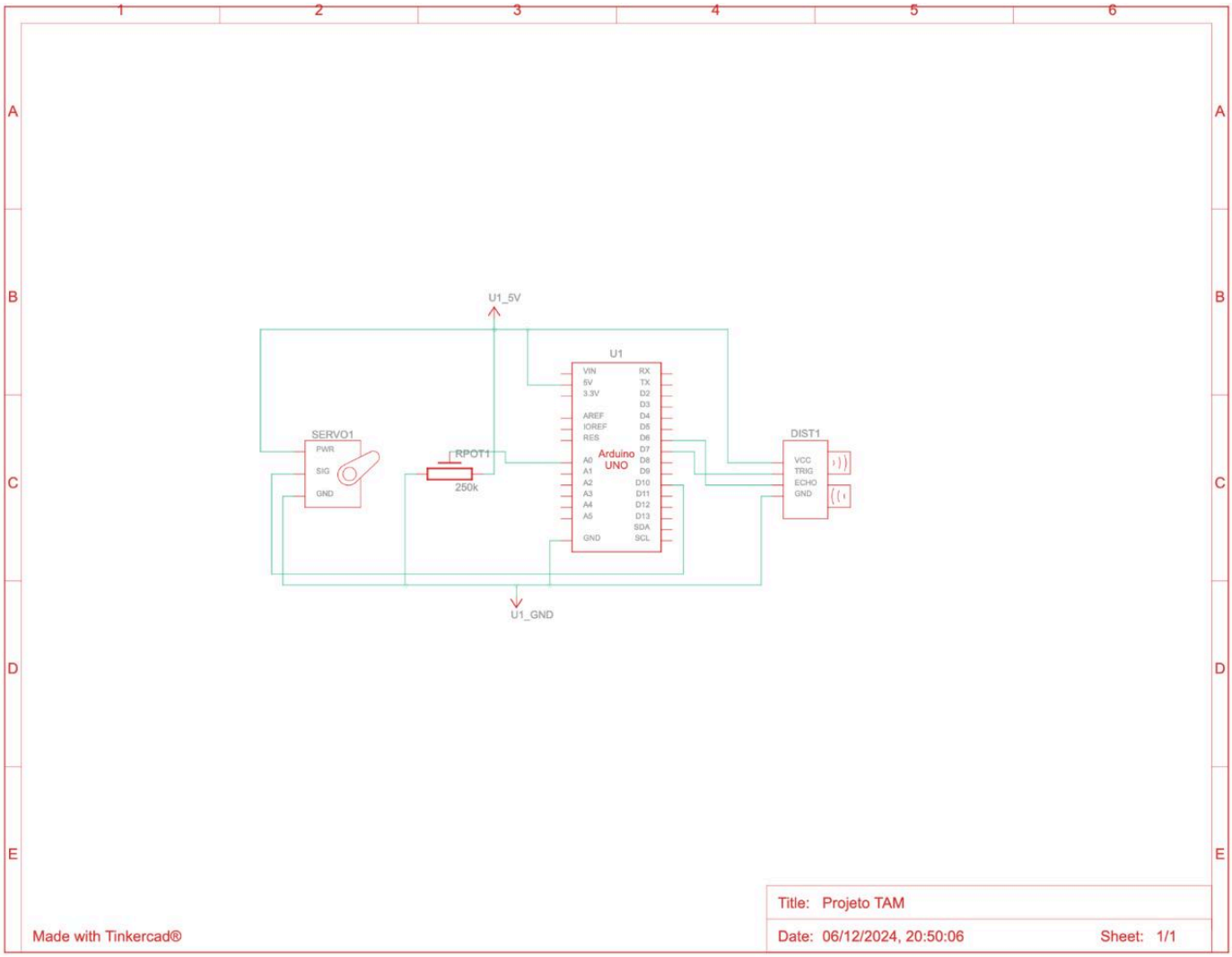

Figure 2: Electrical Circuit Diagram.

- **Power Supply:** Supplies power to the Arduino, ESP32-CAM, and sensors.
- **Ultrasonic Sensor Connection:** Wires the sensor to the Arduino for object detection.
- **ESP32-CAM Communication:** Configures serial communication with the Arduino.
- **Potentiometer Wiring:** Connects to the Arduino for adjusting sensor detection distance.
- **Servo Motor Control:** Linked to the Arduino for gate operation.

The electrical circuit scheme is shown in Figure 2, which illustrates the connections and components used.

Results

The developed system was evaluated under controlled conditions to assess its performance in key aspects such as detection accuracy, recognition success rate, and operational response time. A demonstration video showcasing the system in action, including vehicle detection, license plate recognition, and gate operation, is available for viewing. The video can be accessed via the following link: Video.

Metric	Value
Detection Range	20 cm
Recognition Accuracy	40%
Response Time	0.5s

Discussion

The system shows potential for automating gate access using license plate recognition, offering an affordable and straightforward solution suitable for residential and commercial applications. The use of a potentiometer adds flexibility, allowing the system to adapt to different environments by dynamically adjusting the detection range of the ultrasonic sensor.

However, significant challenges were encountered, particularly in the accuracy of license plate recognition. The system relies solely on traditional image processing techniques without the support of machine learning models, which limited its ability to handle complex scenarios such as poor lighting conditions, partially obscured plates, or damaged license plates.

Future work should focus on integrating machine learning models to address these limitations, enhancing the robustness and accuracy of license plate recognition in diverse conditions. This addition could significantly improve the system's performance and reliability.

Conclusion

This paper presents a cost-effective and scalable solution for gate access control using license plate recognition. The system combines hardware (Arduino, ESP32-CAM, and a potentiometer for range adjustment) with software (Flask API, OpenCV, and Tesseract) to deliver reliable performance. By implementing a validation algorithm based on Portuguese patterns, the solution improves the accuracy of plate recognition. Future enhancements, such as the integration of neural networks, could further optimize the system's functionality.

Acknowledgments

The complete source code and additional resources for this project are available in the public repository: Repository.

References

OpenCV Documentation. https://opencv.org

Tesseract OCR Documentation. https://github.com/tesseract-ocr

entbappy/Car-Number-Plates-Detection. (n.d.). Retrieved December 6, 2024, from https://github.com/entbappy/Car-Number-Plates-Detection/

WiFiNINA - Arduino Reference. (n.d.). Retrieved December 6, 2024, from https://reference.arduino.cc/reference/en/libraries/wifinina/

ESP32-CAM Video Streaming and Face Recognition with Arduino IDE | Random Nerd Tutorials. (n.d.). Retrieved December 6, 2024, from https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/