EYE-PARK

Final Project Report

Authors: Farzam Asad Nejad & Juan Rojas

Photo by Tara Winstead from Pexels

Subject: Technology Project – TPJ655

Program: Computer Engineering Technology

Instructor: B. Shefler, rm. A4022, ext. 26429

Date: 2021-12-03

SENECA COLLEGE OF APPLIED ARTS AND TECHNOLOGY
School of Electronics and Mechanical Engineering Technology
1750 Finch Ave East, Toronto, Ontario, M2J 2X5.
Tel. 416-491-5050 Fax 416-490-9839
www.senecacollege.ca

[This page was intentionally left black]

Table of Contents

Exclusive Summary

In this report we will be providing a technical breakdown of Eye-Park that provides an easy and reliable passive surveillance on private parking garages. We aim to eliminate the hustle involved in facing an unauthorized vehicle being parked in our private parking spots and the frustration in the process of tagging and removing such vehicles. We have spent many hours working on technical features of this system in order to provide a robust build that is accommodated with software support and is easy to implement in already built properties. In the next few pages, we will discuss many different layers of technology involved to build such a system and their specifications.

Introduction

Automation has had a long history of enhancement and development. With the advent of Image Processing and advanced Neural Networks, surveillance systems are much more capable than just recording incidents occurring. Today's technologies allow us to utilize cameras in ways we would never think of years ago. Image processing will transform cameras into all-in-one surveillance solutions that are able to recognize and process the data in their frames. This project aims to provide private parking lots with an advanced surveillance system by utilizing this powerful technology equipped with modern sensors and hardware.

With the development of cities and increase of multi-story accommodations with shared common areas especially in populated metropolitans, the importance of privacy and exclusive access to private properties increases as well. It is apparent that having unauthorized vehicles parked in your private parking lot brings nothing but inconvenience and waste of time.

This project provides a reliable monitoring system that deploys cameras backed with image processing to capture and read plate numbers of vehicles entering a specific spot and compare the number to a database of registered vehicles and in the case of no match, it will proceed further actions based on user input via a companion app.

Functional Features of the Product

The system consists of a box installed at a reaching point on a wall or a pole at the end of the parking spot with a camera housed inside. It also has an LCD screen, an ultrasound proximity sensor, as well as a keypad and two LEDs. Upon entry of a vehicle, the proximity sensor will recognize its entrance and will trigger the camera to exit its state of standby. The camera will then read the content of the vehicle's plate number by using the OpenALPR library. The vehicle’s number will then be compared to a list of previously registered vehicles. If the vehicle has been registered, a message will be sent to the owner via the companion app. Then the LCD screen installed on the spot will show a “vehicle authorized” message and will turn the green LED on.

If the vehicle’s number is no match to the database, and the vehicle stays for more than two minutes, the system will notify the user via the companion app. After showing the vehicle's information and time stamps, it will allow the user to request tow/tag service from security or to add the number to the list of registered vehicles or to dismiss the alert.

At the case of unregistered vehicle entrance, a red LED will turn on, an appropriate message will appear on the LCD screen that will prompt the user to key in a 4-digit user pin that will add the license to the registered vehicles’ database and will disengage the system and will turn the green LED on. If there are more than three wrong attempts of a user pin, the system will require admin code to unlock. This physical approach is installed as a backup for times that access to the app or internet is not possible. Note that the companion app and the physical approach of using admin PIN work side by side and are in synchronization with each other.

Specifications of the Product

• Product dimensions: 25cmx10cmx7cm

• Display type: I2C LCD 1602

• Distance sensor: Ultrasonic Ranging Module

• Image Processing Library: OpenALPI

• Camera resolution: low light 1080P

• RGB LED to provide status of vehicle

• Entry distance: 3m

• Unauthorized vehicle’s max time to stay: 5 minutes

• 4x4 matrix keypad to add new entry to the database

• Power supply: 5V — 2.5A Micro USB Power Supply with Noise Filter

Operating Instructions

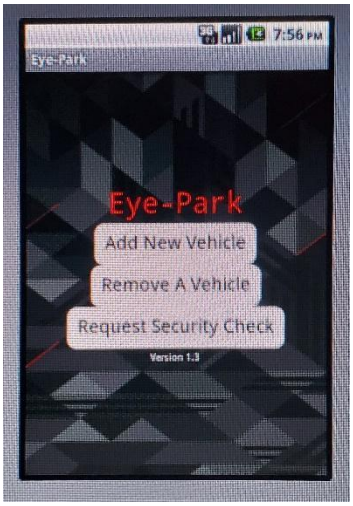
When a car is parked in the assigned parking spot, an LED light will indicate if the car has been recognized as an authorized vehicle or not. If the device indicates with a red light that the car is not recognized, it will send an alert to the app and will display a prompt. Either of them can be used to authorize a new car. If the app is used it will give the options of authorizing the vehicle or to alert security. If the physical interface is used, a master pin must be entered to authorize the car. Security will be alerted after 10 minutes if not authorized.

Manual for unauthorized guest vehicles with access code:

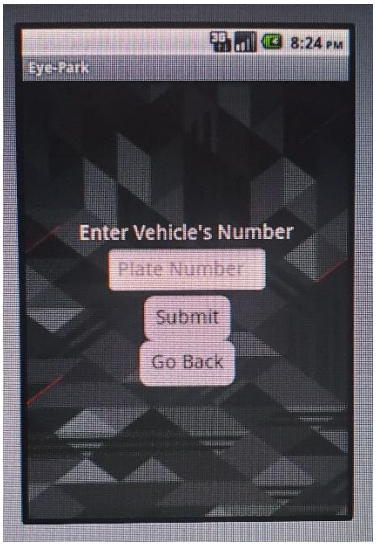
1. Enter the parking spot and park your vehicle
2. Approach the screen and confirm your plate’s number printed on the LCD screen
3. Enter the 4-digit access code then press “#”
4. Watch the Green LED turn on
   Note: For guests without access code please contact owner and they will provide it for you.

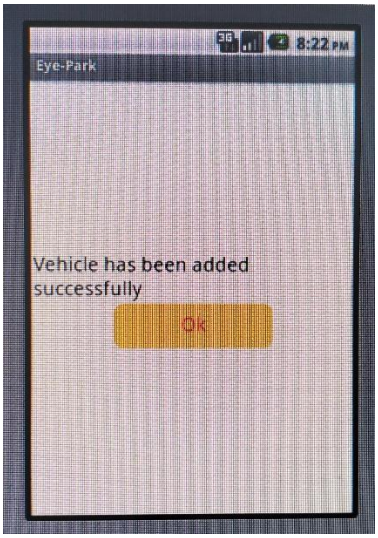
Manual for Eye-Park Android App:

1. Download the app and install it
2. Open the app
3. First page will provide you with three options

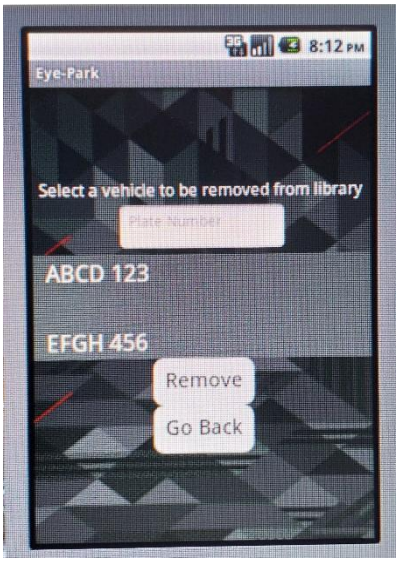


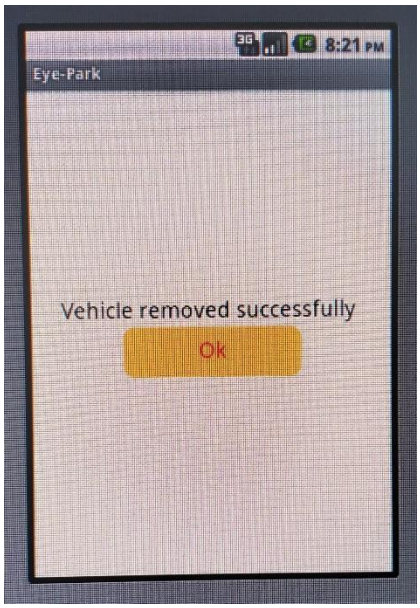
4. Press “Add New Vehicle” to enter a vehicle’s plate number
    i. Press “Submit” to send the data to the server or “Go Back” to be redirected to the main menu


8:24 PM
Eye-Park
Enter Vehicle's Number
Plate Number
Submit
Go Back

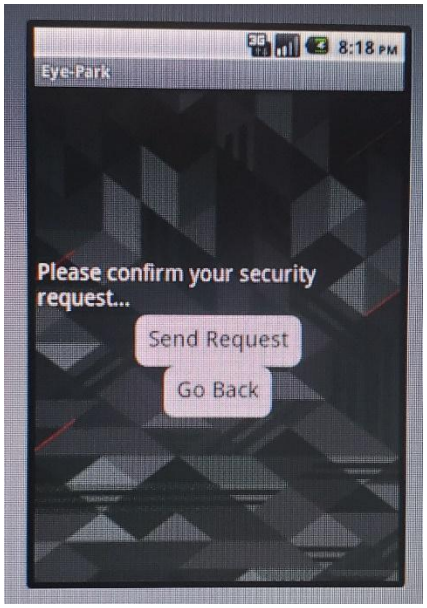

8:22 PM
Eye-Park
Vehicle has been added successfully
Ok

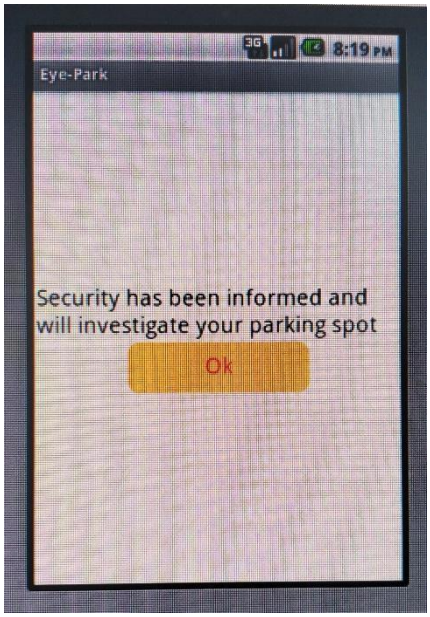
5. Press “Remove A Vehicle” to delete a specific number from the database
    i. Select a desired number from the list of existing vehicles in the database or enter the plate number manually
    ii. Press “Remove” to apply changes to the database or “Go Back” to be redirected to the main menu





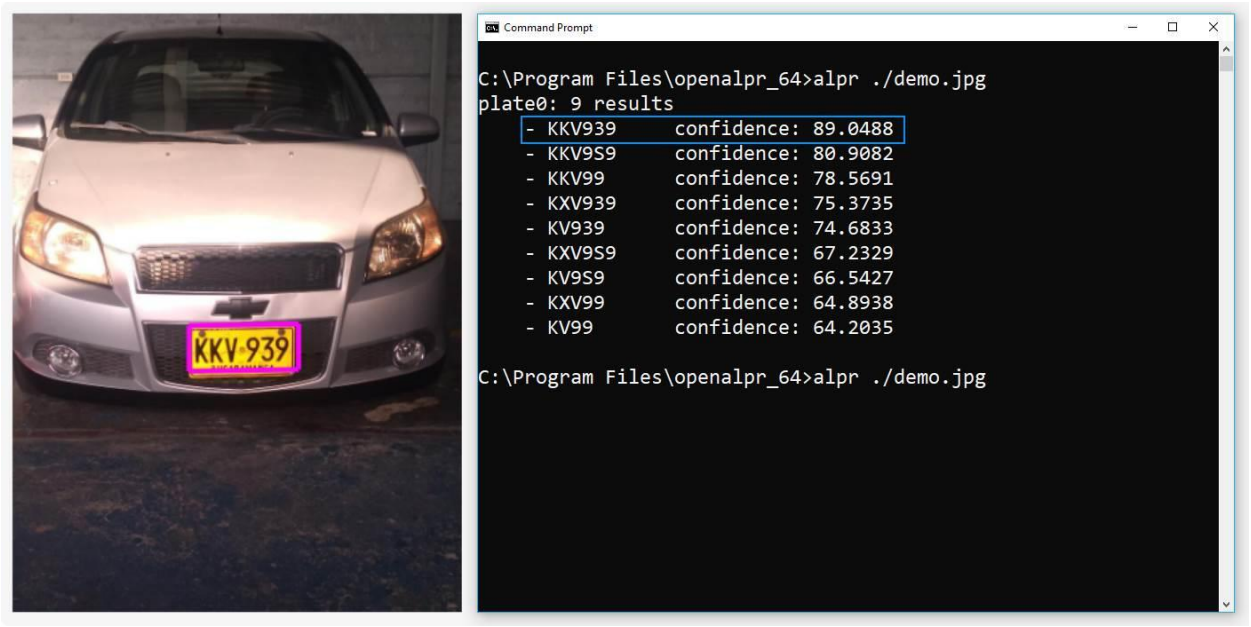
6. Press “Request Security Check” to request a patrol by concierge
    i. Press “Send Request” to confirm your decision or “Go Back” to be redirected to the main menu


8:18 PM
Eye-Park
Please confirm your security request...
Send Request
Go Back

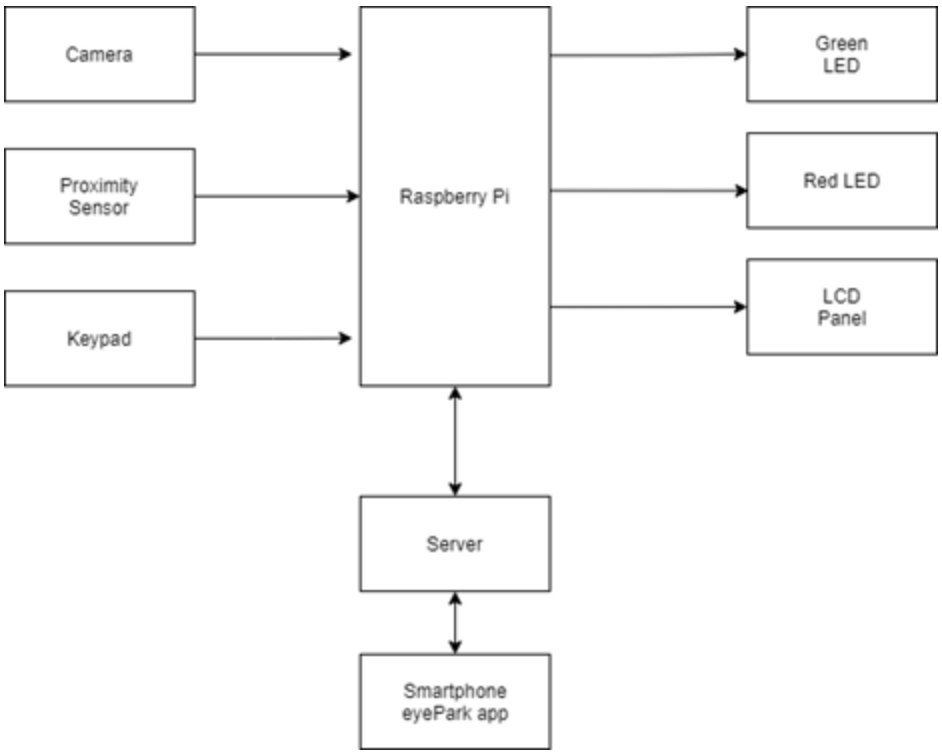

8:19 PM
Eye-Park
Security has been informed and will investigate your parking spot
Ok

Product Design, Implementation and Operating of the System

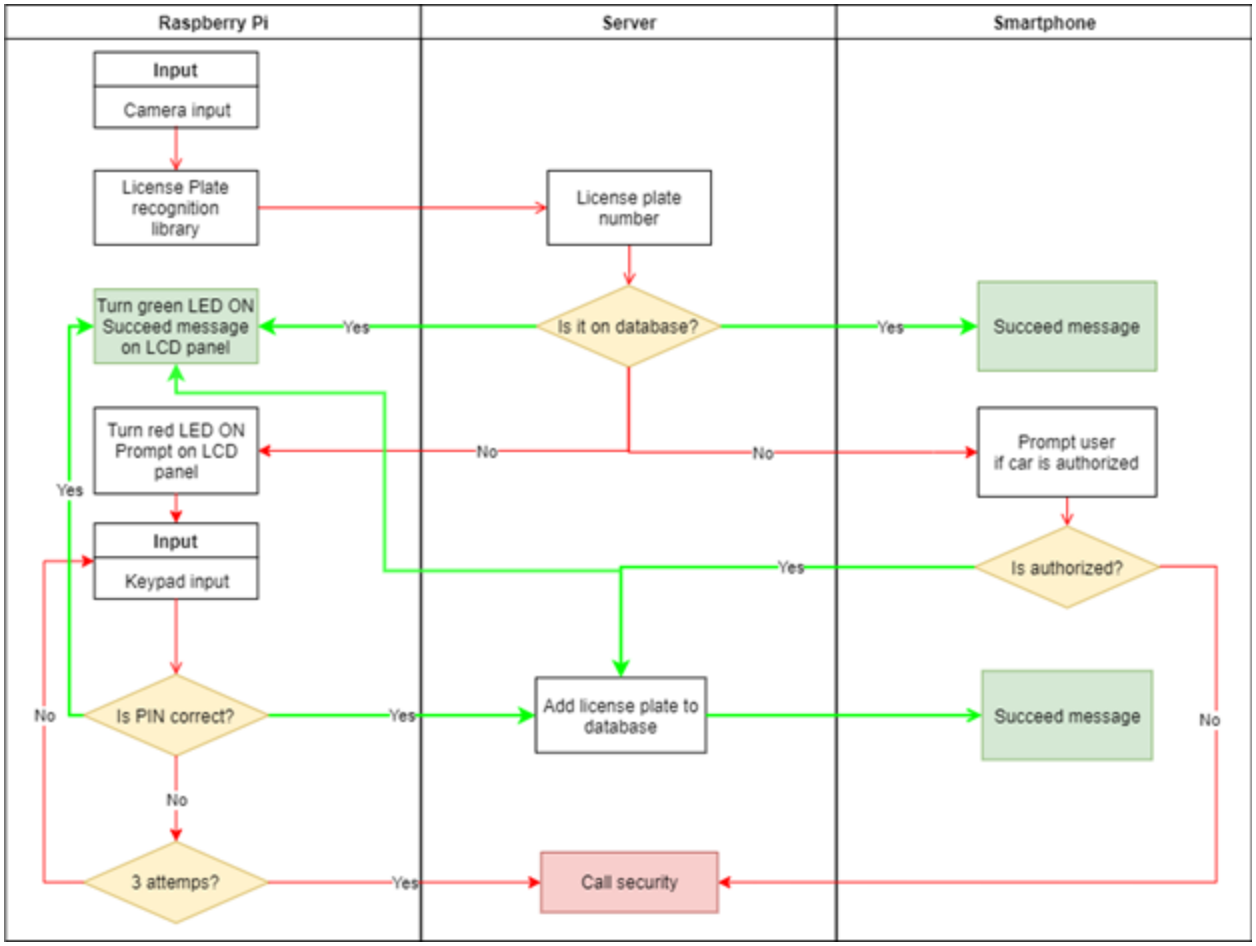
Sample CLI run of OpenALPI



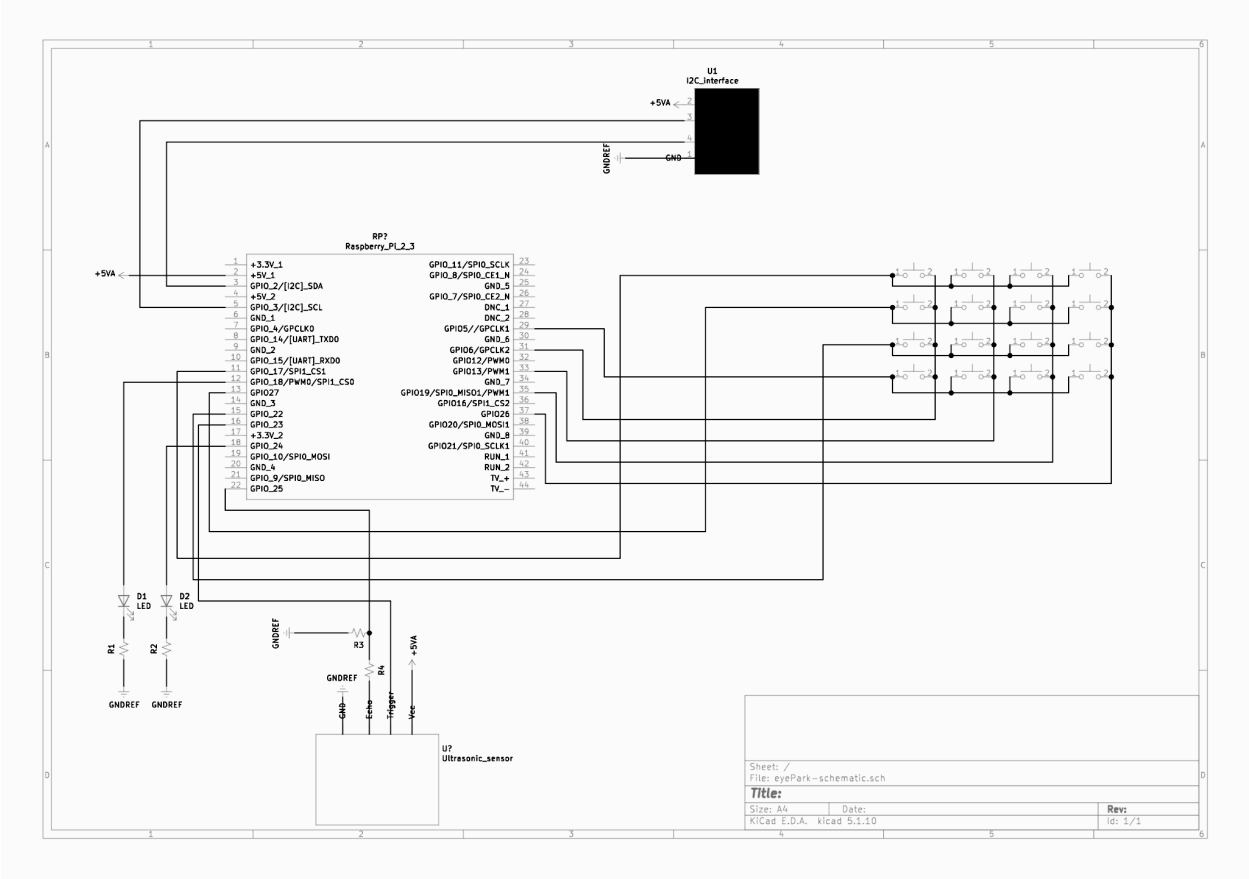
System block diagram



Flow-chart


Raspberry Pi
Server
Smartphone
Input
Camera input
License Plate recognition library
License plate number
Turn green LED ON Succeed message on LCD panel
Yes
Is it on database?
Yes
Succeed message
Turn red LED ON Prompt on LCD panel
No
No
Prompt user if car is authorized
Yes
Input
Keypad input
Yes
Is authorized?
No
Is PIN correct?
Yes
Add license plate to database
Succeed message
No
No
3 attemps?
Yes
Call security

Circuit diagram



Main Components Featured

Raspberry Pi 3 B+

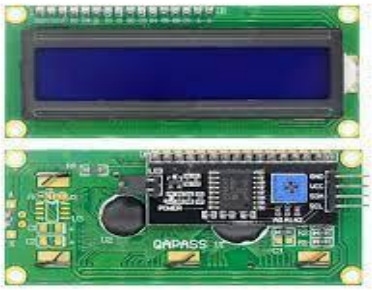
The development board chosen for this project is a Raspberry Pi 3B+. This board has been proven for its simplicity and integrated features making it an ideal board for developers and engineers all across the world. The Cortex-A53 64-bit processor of this board operates at 1.4 GHz which can provide an adequate processing power for the openALPR library which can sometimes be resource dependent. The processor is accompanied by a 1GB DDR2 memory which can provide an adequate memory for this design. In addition to its processing power, the board provides a reliable Ethernet connection as well as wireless connectivity that allows for a more simple and clean connection to the board.

This board provides an extended 40-pin GPIO headers that allows for multiple components to be connected to without any issues. Overall, the features of this board makes it a great candidate to be used in this system in the initial stages.

Camera

This High-Resolution camera provides a detailed image at 4K with pictures as sharp as 1080p in low-light situations. This low-light capability allows the system to capture frames in not so ideal lighting situations such as underground and during night. The camera communicates with the Raspberry Pi with a standard USB type-A. Another smart feature of this camera is its low power consumption and its smart way of being on standby when not being used.

I2C LCD Screen



The i2c LCD screen is a reliable component that is designed for simplicity. The screen can be connected to the Raspberry Pi via reduced 4-GPIO pins. This screen is ideal due to its low power consumption and high visibility. The screen

is made out of rigid materials and is ideal to be used in polluted environments such as parking.

Proximity Sensor

This Ultrasonic Ranging Module provides a 2cm - 400cm range of measurement. This module is made of ultrasonic transmitters and receivers. This module can operate at an angle of 15 degrees that sends eight 40 KHz pulses and calculates the time of receiving pulses which can provide the distance of the reflecting object. This sensor operates on a 5V supply which makes it a great component for this low-power consuming device.

Theory of operation

Server

The server uses the Django python framework to receive and respond to requests. Django uses the MVT (Model-View-Template) model which allows for a very scalable design. The model interacts with the database, Sqlite in this case. The view sends requests to the model and modifies the templates to respond to a request. The eyePark server doesn't use templates at the moment, as the responses are simple HTTP codes and not any HTML pages. For handling databases Django uses ORM (object-relational mapper), this means that the databases can be created from classes set up in the models, without the need of writing any SQL. This functionality also allows an easy migration to different types of databases without too much hassle.

Scanning license plate

The library used is called OpenALPR. This library uses OpenCV for computer vision and Tesseract for OCR (optical character recognition). OpenALPR has the ability of recognizing the characters on a license plate and giving a percentage of confidence that the recognized pattern is the correct one. This data can be accessed either by a binding in different languages, or by using a daemon that can submit the data to a beanstalkd queue, which can later on be processed and submitted to the eyePark server.

Send request to check plate

When a license plate is read, the program will send a GET request to the eyePark server. The request contains the number of the parking spot, the license number and a unique key. The unique key is assigned at the time of deployment and prevents a malicious user from making requests to the server if they are not authorized.

Send request to add new plate

If the server responds to the check plate request with a HTTP status code of 404, meaning not found, and the user authorizes the vehicle to be added; either by entering the master PIN, or by accepting in the app. The device will then send a POST request to the server that has the parking spot number, the license plate number and the unique key. The server will return a HTTP code of 200 if the request was successful.

Send notification to security

If the system checks with the server and the license plate is not in the database and the user either doesn’t input the PIN in the keypad or accept through the app, the Raspberry Pi will send a POST request to the server that contains the parking spot number and the license plate number as a JSON file and the server will add it to the SecurityAlerts database. Security has access to the Django admin page and only has access to the data for the security alerts database. The SecurityAlerts database has information on the parking spot number and the license plate. Security also has the ability to delete an entry on the database when an appropriate action has been taken.

Server-client communication

As mentioned before the raspberry pi client is capable of sending GET and POST requests to the server in order to do the different actions that it needs to do. The client sends requests to an url that was setup in the configuration file. This url is not of a local ip, but rather of a DNS record. This record was obtained with the no-ip service, which allows for a Dynamic DNS or DDNS. Since a residential internet service provides only dynamic ip's, unless the customer pays a fee to get a static ip. The DDNS service is imperative to this application since the IP would have to be updated constantly in the DNS server.

OpenAlpr issues

While the algorithm is really good at reading license plates it can have some troubles in reading a plate correctly when encountering different lighting conditions, when the vehicle is in motion or different angles. To deal with the majority of these issues the system loops up to 10 times reading a plate until a correct plate is recognized, giving the system more chances to get a correct reading.

Maintenance Requirements

Due to the active monitoring feature of this project, regular inspections of connectivity of the system as well as status of its camera and processing units are needed. In addition to physical upkeep of the equipment, the server will need monitoring and backup service on a regular basis. Considering the fact that this project is heavily relied upon consuming power for long processing times, it is important to keep the system's housing box clean of any dust and to inspect the status of ventilation fans and other air intakes. Lastly it is advisable to keep the camera's lens clean of any visually obstructing stains which have a tendency to absorb more dust and fumes emitted by surrounding vehicles.

Further Developments

For future developments the camera can be used for a certain area of the parking lot, instead of one per spot. It can also provide functionality that allows communication to the police database, monitor and report back any car that was reported as stolen or suspicious. This project can also be used while capturing images via heat sensitive

infrared cameras that could provide excellent top-notch security surveillance or used in SAR (search and rescue) missions by highlighting hotspots of any given frame.

The project will be updated using GitHub, the same way that the project was developed. The Eye-Park repository can be found at: https://github.com/jrojas-pena/eyePark

Conclusion

Overall, the results of tests done are extremely promising of a future where, without the need for expensive equipment and technologies, AI could be utilized in homes and provide excellent surveillance capabilities, as well as many other helpful ways. It is also mentionable that with utilizing graphic specific processors in small form factors the speed of processing each frame of a video or pictures can be increased to provide a better and more efficient service. After weeks of research, development and deployment, Eye-Park proves to be an extremely powerful tool that can be used in many industries therefore aiding humans in building a more advanced future.

References

https://www.djangoproject.com/
https://github.com/openalpr/openalpr
https://www.raspberrypi-spy.co.uk/2015/05/using-an-i2c-enabled-lcd-screen-with-the-raspberry-pi/
https://maker.pro/raspberry-pi/tutorial/how-to-use-a-keypad-with-a-raspberry-pi-4
https://appinventor.mit.edu/

Contact Information:

	Student name	Email	Phone number
A	**Farzam Asad Nejad**	**Fasad-nejad@myseneca.ca**	**+16476777720**
B	**Juan Rojas Pena**	**Jrojas-pena@myseneca.ca**	**+16479147216**