Team number: 18

Team members: Ahsan Muhammad

Bikash Kalikote

Huzaifa Zahid

Mir Shazil

RokunuzJahan Rudro

Project name: *ParkMyCar*

Project Synopsis:

An app that displays the total number of available parking spaces in the selected building and total number of spaces available on each floor. Additional features include: permit information, lot timings, game-day alerts, campus lots listing with respective permits.

1. Introduction

Our goal is to build an app that utilises image recognition to detect whether a car is leaving or entering a building. Based on that, we will predict the number of spaces available in the building and on the selected floor. The accuracy will be less than 100% since one or two cameras will be used per level, at the entrance/exit.

2. Purpose

We are going to try to minimise the hassle for the user to find a parking spot. Users go into a parking area and then have to look around for empty parking spots. A lot of the time, it turns out that the parking spaces on the floors are already taken and the user just has to turn back and go to the next floor. Our app will use computer vision to see for cars and number plates to check if a parking space is occupied or not

and then populate the app with the information for the user. Many people sometimes come to a parking lot and then keep finding a spot without knowing if there is any spot left. This sometimes creates a traffic jam in the area because the lot gets populated with cars. This can be avoided if these people already knew that there are no spots left which is the purpose of our application. Additionally, our app would have information about all the parking permits and would show which permit allows access to which parking lots and if at any day access isn't granted, like on game days. This would be a helpful feature for users who are planning on buying a parking permit, but don't know which permit would be better for them, in terms of the lots they get access to.

3. Main Body

3.1 Front- End:

Front end will mainly consist of user interface component which will have the following features:

a. Menu

The menu component will show us all of the options regarding the application that a user can choose.

i) Parking Lot:

The user will be redirected to a list of on-campus lots. From there, the user can select the lot and will get redirected to a page with the number of available parking spots per level. This will also include the timings for the parking lot. This will be useful for finding an available spot without spending too much time trying to find one.

ii) Permit:

Users will be redirected to a list of campus parking permits. From there, the user can select a permit, and get redirected to a list of lots they can park in with that permit. This will be useful to find lots the user can park on with their current permit or help in the decision making process of choosing a permit.

iii) Explore Lots:

The user will be redirected to a table of campus lots and the permits allowed. This will be useful in getting an overview of all the lots on campus.

b. Buttons:

As discussed earlier, the main screen will have a menu with all the options so for that, we will need to have virtual buttons on the screen placed systematically so that a user can tap on them to proceed. There will be a start screen which will lead to an options screen with buttons for Parking lot, permit, explore permits, and settings.

c. Display:

Firstly, we will use frameworks or plugins that will make the application's screen fit on any phone's screen. We will have a splash screen, showing the logo of our application to familiarise people with our app. Then we will have the Home screen, which will list the menu options, idea of our product, and various features of the application.

3.2 Back- End:

The app will have a critical support system. Firstly, we will have a machine learning model which will be trained to identify which objects are vehicles and which are not. We will create a convolutional neural network and feed it with datasets containing images of backside

of cars and front side of cars to let the model train itself and get firm on how a car looks from both sides. Then we will provide the model with training sets to make sure that it has attained the expected accuracy. We will use multiple python libraries to use effective functions etc. to make our neural network work.

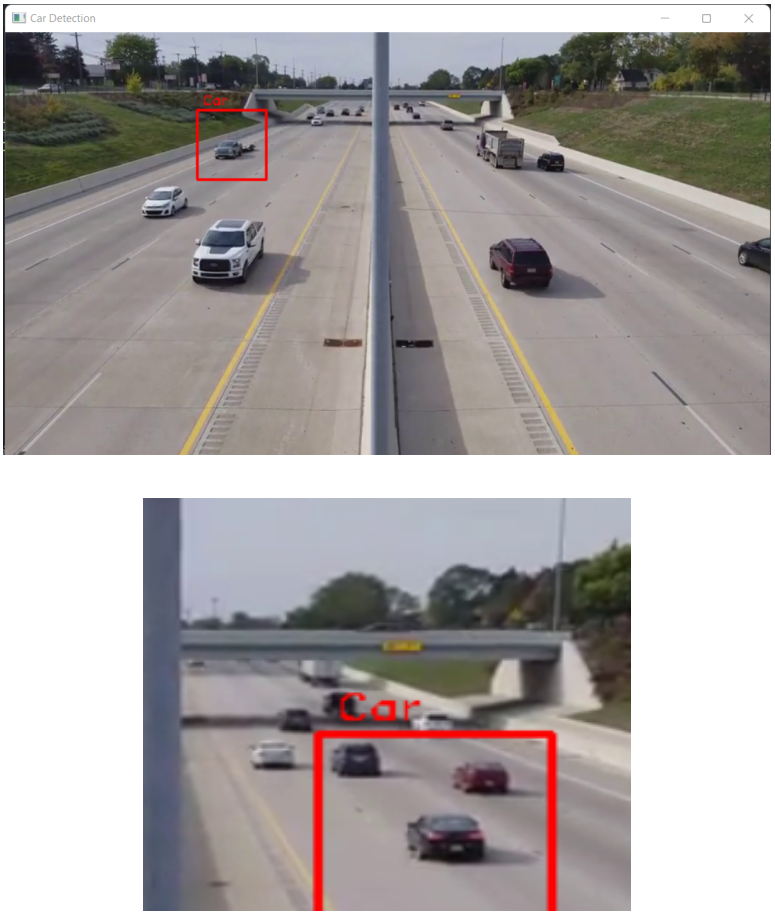
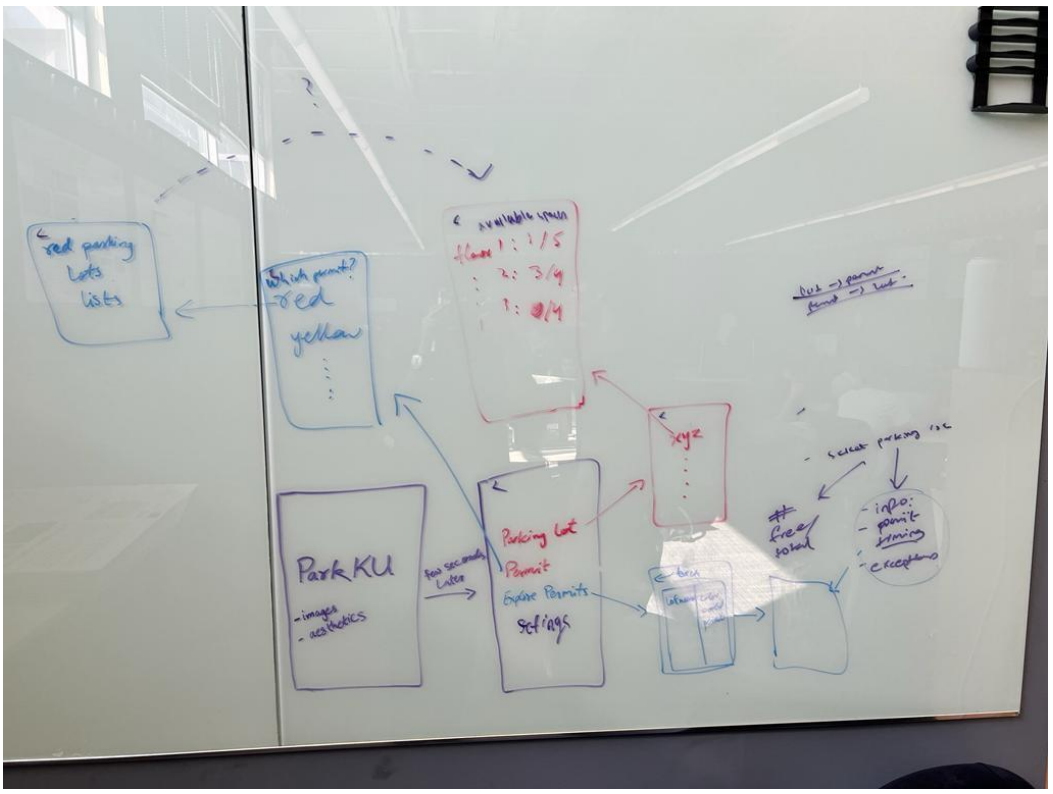


Fig: detecting cars using **openCV** library

For the mobile application's development, we will need to use a framework to create it. We will be using Flutter framework, which is an open-source, cross-platform app development framework owned by Google. Flutter uses Dart, an object oriented programming language that is client optimised language to provide a high-performance user experience, and is used for creating native cross-platform apps for IOS, Android, and Windows. The following diagram shows the tentative flow of our app:



As our application will have additional features like showing which parking lots are unavailable for parking even with a parking permit on certain days like game days, we will need to know those dates. For that, we will use a database to pull off the data from a calendar and store it into it. This way, users will know in advance which parking lots will allow them to park and which won't.

For the hardware part, we will have Automated licence plate readers (ALPRs). ALPRs are high-speed, computer-controlled camera systems that are typically mounted on street poles, streetlights, highway overpasses, mobile trailers, or attached to police squad cars. We can use them too because this way, we can provide more accurate results and not mistake a person for a car. ALPRs automatically capture all licence plate numbers that come into view, along with the location, date, and time. The data, which includes photographs of the vehicle and sometimes its driver and passengers, is then uploaded to a central server.

4. Project Constraints:

One of the major challenges for the project would be to connect the machine learning model with the camera and then transmit the data between all of the components of our mobile application. We will need to make sure that our transmitting medium doesn't delay the data exchange (sending and receiving) because this will cause the available spots to be inaccurate. Another issue would be to incorporate the database into our application model to satisfy the condition of additional features.

5. Ethical issues

There are some ethical issues concerning users. One of them is privacy. Of course, there will be people walking in and out of the parking lot. Since we are using cameras to identify vehicles, cameras will capture these people too which means that they will be in the video or even picture without them knowing about it. Also, these cameras can just make people uncomfortable if they are too many and if they are very much evident as if they are following the people on foot. Another issue could be that on a large scale, there might be many parking lots that do not give us the permission to install cameras, claiming that we might be spying on someone or that we might be trying to identify someone in specific.