Chapter 1 - Introduction

1.1 Introduction

The bulk of people reside in cities, and the population of the world is increasing every day. More than six billion people will inhabit the globe by 2045 (I. M. Hakim, D. Christover and A. M. Jaya Marindra, 2019). People cannot deny that cars are a part of their daily life, and it seems that cars are essential in today's culture (Koushika, B., PraveenKumar, G., Priyadharshini, S.T. and Balamurugan, A., 2021). Parking in cities has become a major issue as the number of cars on the road increases. Lack of real-time parking spot monitoring, particularly in congested areas, will lead to vehicles searching for available parking spaces, parking in risky locations, or parking in a parking area reserved for people with disabilities (M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, 2018). Therefore, a smart parking system with automatic vehicle identification is required to give car owners rapid and simple access to accessible parking places.

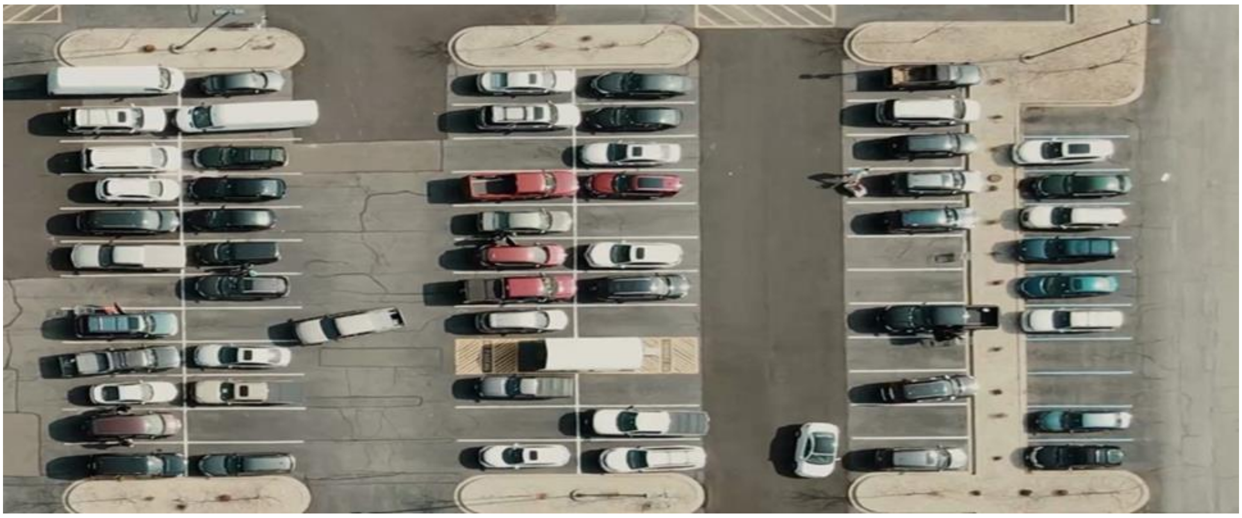


Figure 1: Parking space detection using digital image processing

Figure 1 shows a snap of the sample that will be used for our project. We aim to reduce the cost of the system by using minimal hardware and minimizing the search time for the parking spot. Authorized customers can get parking information from the real-time firebase database. The proposed system will utilize Python and the OpenCV library to detect of a parking spot is occupied or not from live video. Cameras were placed in strategic spots to detect ticketless parking. Authorized customers can get parking information from the real-time firebase database.

1.2 Problem Definition

Smart Parking is a method for parking cars that aid drivers in finding a free space. In today's culture, the number of vehicles is constantly increasing. Finding a parking spot is a big problem since the number of automobiles is growing so quickly, whether at workplaces, shopping centres, airports, or medical facility finding a parking spot for one's car usually take around 20 minutes. Most people leave their cars parked in positions that are not designated as parking spaces and in regions that are not marked as such. disruption in both the movement of vehicles and occasionally humans. This is due to poor planning, a lack of innovative problem-solving, and the fact that most people park their cars for periods without thinking. Since parking is a frequent and ongoing issue, a solution is required to minimize traffic congestion and air pollution at least partially, as it is a common and recurring problem.

1.3 Aim & Objectives

1.3.1 Aim

This project will create a smart parking system with digital image processing. The total number of parking spots will be updated to the live database after the car is parked in the parking lot. A camera will be employed at the entrance point to record the number plate of the automobile and add the car number to the live database. One more camera module will be installed at the exit point, and it will once again read the vehicle’s license plate, delete the vehicle's number from the database, and update the total number of parking spaces available.

1.3.2 Objectives

- The primary objective is to cut down on the time it takes to find a parking spot and the stress that comes with it.
- To correctly guide a driver to a free parking spot lowering CO2 emissions, noise levels, and other contaminants.

- Minimize lingering and excessive driving, and hence improve traffic flow in congested regions.
- The created prototype is portable, it has been examined in Results & Analysis.
- To develop new business models based on technology, such as app-based or tag-based payments, real-time and electronic payments, and dynamic parking.
- To adopt Smart City technology or standards like ISO 37122.

1.4 Tasks

The system formulation and construction, as well as the final report, are the two key duties in this project. The following tasks are covered in the system design and implementation section:

1. Configuration of the system, image sensor deployment, and image processing.
2. Image identification (ROI: A region of interest (ROI) is a segment of a picture that you wish to analyse or manipulate).
3. Image acquisition (The goal of any image acquisition is to convert a visual picture (real-world data) into a quantitative data array that can then be edited on a machine).
4. Image conversion (binary conversion).
5. Image Enhancement (remove noise from the image).
6. Detect the number of the free parking lot.
7. Upload data on Firebase Realtime Database.
8. Get data and display it on the user application.
9. Test all the modules in depth.

1.5 Hardware Required

- Mobile Camera modules will be installed on the entry and exit points on the prototype developed.
- Toy Car will be used to demonstrate the occupied parking & the available space in a parking lot.
- A3 size cardboard is used to create a prototype parking area & marking of parking spaces is done according to the toy car.

- A tripod camera mount is used to mount the camera on the entry & exit points.

1.6 Project Risks

Prioritizing risk analysis will help to lessen the likelihood of failure. There is a considerable probability that the project won't be finished because of how quickly it must be finished. The timeframes listed in this report are therefore presumptive, and there is a possibility that they might be missed. As a result, the extra time has been provided in each subsection section to make sure that no subsection's time limit is missed.

Failure of the database or hardware failure and the time needed to replace it might both pose technical risks to the project.

1.7 Gantt Chart

A Gantt chart depicting the distribution of all project activities is shown in ***Figure 2***. To show the whole output of the solution, all these tasks must be finished. There are nine parts to the project. While some have already been finished, others will be shortly. While some of the tasks are connected to one another, some can be completed on their own. Here is an illustration of a Gantt chart.

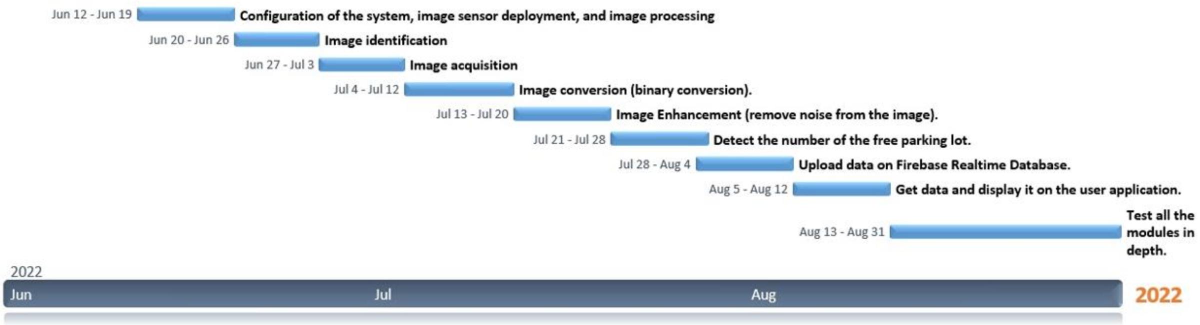


Figure 2: Gantt Chart.

1.8 Organization of The Dissertation Report

The format of the project dissertation report is as follows:

Chapter 1: Introduction

This chapter provides an overview of our suggested system, as well as the problem definition, goal, and purpose that this system is intended to achieve.

Chapter 2: Literature Survey

The Literature Survey, which includes several current systems and the strategies employed in them, is covered in this chapter. It also includes a comparison chart that lists the contributions, strategy, tools, and overall limits of several methodologies.

Chapter 3: Design Methodology & Implementation

This chapter provides details on the system's general design, operation, and implementation, outlining each phase in great depth. The project's flow is included.

Chapter 4: Results and Analysis

This chapter analyses the findings and contrasts the recently implemented system with earlier efforts.

Chapter 5: Conclusion and Future Work

This chapter finishes by explaining how the recently introduced system solves the shortcomings of earlier initiatives and outlines potential future improvements that might be made to the system to increase its effectiveness and dependability.

Chapter 2 – Literature Survey

2.1 Introduction

The use of digital image processing in smart parking systems has been the subject of many studies, which are done in this part. This chapter offers a literature review of several technical publications, as well as information on the benefits and drawbacks of each system.

2.2 Literature Survey

2.2.1 Intelligent Recommender System for Smart City Parking Vehicles (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021)

Finding a parking spot nearby is a must for every automobile trip, hence parking is an essential component of settlement transportation systems. Such problems may be prevented by making good use of information, doing studies on the subject, and constructing a parking lot with a conceptual plan. Before informing a customer of the unavoidably available parking spaces in each area, the built-in system examines a particular region, population, automotive ownership rate, location availability, employment characteristics, and other data (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021).

Parking for public transit is one of the largest problems in many cities, and it's usually the result of bad planning. The capacity to efficiently manage traffic and take into consideration parking demands in the city should be at the centre of any future IT programme for a city aimed at enhancing planning in these areas. This study looked at an IT project to develop an intelligent parking system for smart cities (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021).

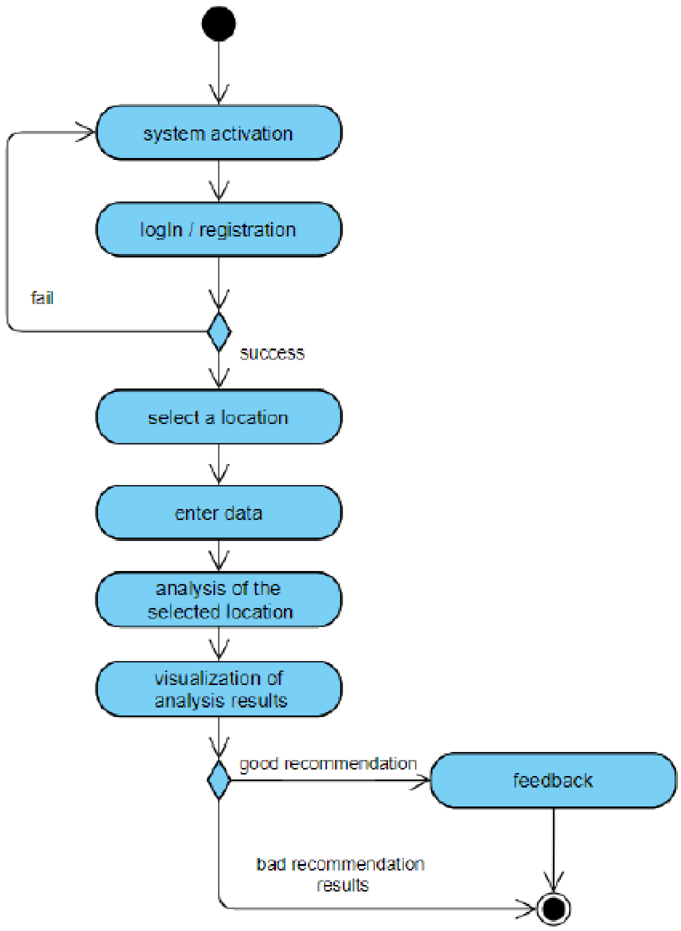


Figure 3: Intelligent recommender system for parking vehicles activity diagram.

(Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021)

Figure 4: Information output page Intelligent recommender system.

(Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021)

Users will be able to ascertain the number of parking spaces available by providing their location and other personal information. ***Figure 4*** This method is intended for anyone interested in figuring out how many parking spaces are required in a modern metropolis. (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021).

The user and server will have the best communication while using the public transportation system to access parking information. Now, the user fills out a question-and-answer form, which is then uploaded to the server. The server processes the data, and the user is given the findings for additional analysis. ***Figure 4*** They can also complete a form with text boxes for unanswered questions and evaluation spaces for the outcome. (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021).

The attempt to build an insightful recommendation system is beneficial with urban transportation parking solutions in smart cities. By picking a place and entering the essential personal information, the user may obtain an answer in a very short period. This method has the potential to successfully address urban expansion and new transportation route planning difficulties. The tool will enable the examination of planned constructions as well as the comparison of proposed and expected parking spaces (Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, 2021).

2.2.2 Intelligent Recommender System for Parking Vehicles in Smart Cities (Shaaban, K. and Tounsi, H 2021)

An innovative method for automatically identifying available parking places was presented in the research. The suggested method uses grayscale photographs taken with any camcorder to identify open parking spaces. The method was tested using real data, and the results showed a 99.7% detection accuracy. Real-time parking availability tracking and vehicle routing could both be done with this system (Shaaban, K. and Tounsi, H., 2021).

Several factors affect parking availability and driver satisfaction in the development of smart cities. In smart cities, parking is a crucial component of the infrastructure. Automated parking identification systems can help with this, which is good for the environment, parking lot owners, and drivers. This study's objective is to develop a new parking space detection tool that can determine when a parking space is occupied (Shaaban, K. and Tounsi, H., 2021).

Our earlier research, which produced two more parking recognition algorithms and was summarised in the review publication, serves as the foundation for this investigation. However, there is always a need to find new technologies that are more accessible, more efficient, and more accurate. The method described in this article calls for a little Several or perhaps just one video camera may be employed, depending on the size, shape, and location of the parking lot. Finally, the data provided by a detecting system need to be accurate. saving time and money for parking lot owners and users, as well as reducing travel time for the environment, pollutants, and the amount of gasoline used by vehicles looking for parking spaces (Shaaban, K. and Tounsi, H., 2021).

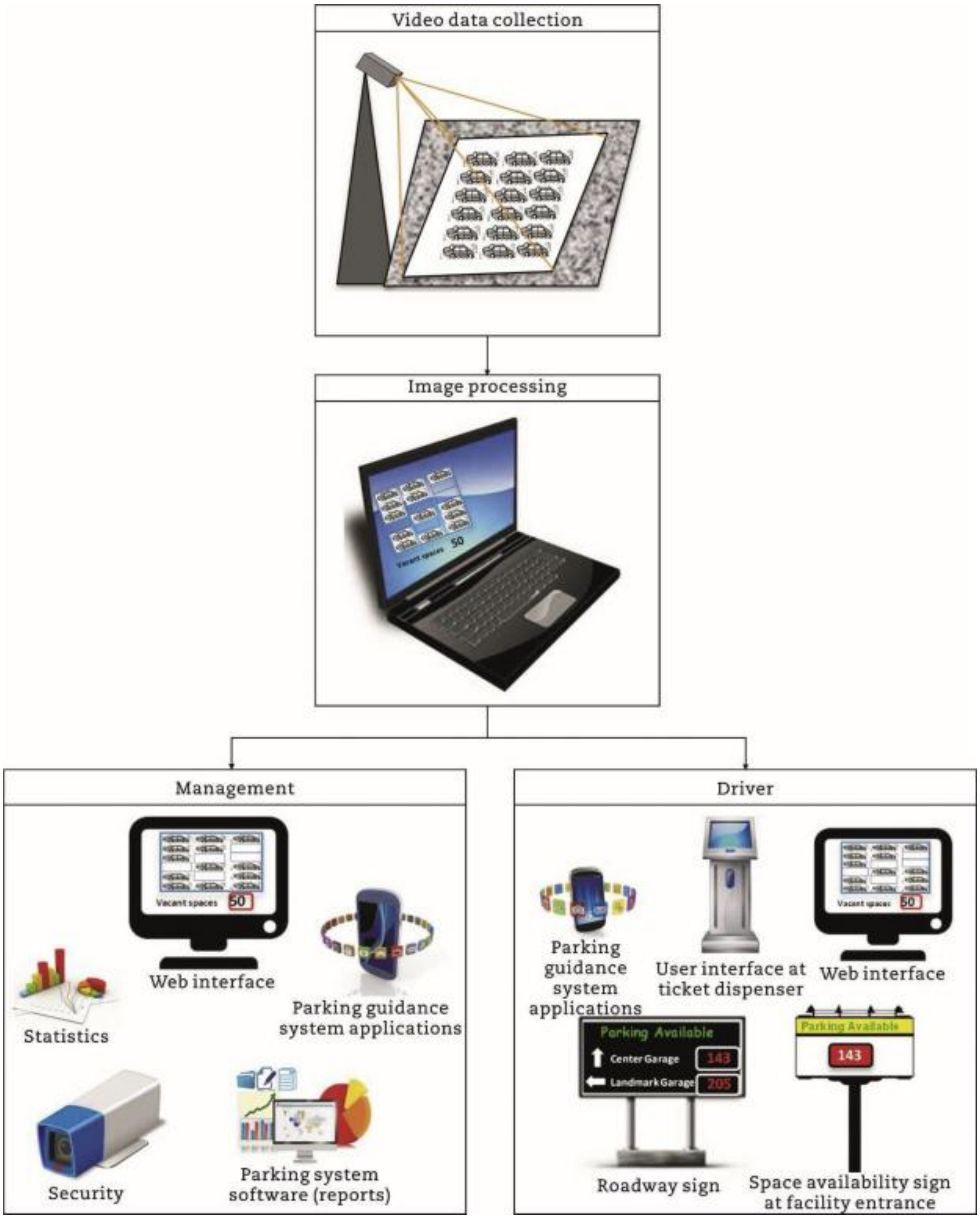


Figure 5: Smart Cities Intelligent Recommender System for Parking Vehicles

(Shaaban, K. and Tounsi, H., 2021)

The recommended approach is photos obtained from video cameras positioned at a high altitude to provide coverage for the required region. To begin the processing, the videos are transferred to a central server. The suggested approach is then used to offer real-time information on the quantity and location of available parking spaces. This data is delivered to drivers via a variety of channels, including local dynamic message signs or at the entrance, a mobile app, a menu system at the ticket terminal, a web-based interface, and so on. ***Figure 5*** displays the proposed system architecture (Shaaban, K. and Tounsi, H., 2021).

The increasing population of most cities require the establishment of effective transportation systems and long-term transportation systems that take full use of modern technologies. Smart parking systems are a critical component of smart city infrastructure. The study established a novel technique for finding available parking spaces on outdoor lots using grayscale pictures by video cameras. The recommended approach is simple and accurate in detecting parking slots that are unoccupied in a variety of circumstances. The suggested method obtained a detection accuracy of 99.7 % using factual data (Shaaban, K. and Tounsi, H., 2021).

2.2.3 Car Detection in Roadside Parking for Smart Parking System Based on Image Processing (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

To employ an intelligent system as a parking administration tool and to learn about available parking spots, this project intends to identify cars parked outside of a parking garage. The Haar Cascade Classifier and YOLOv3 were used by the researchers of this study, and they compared them to see which one had the best accuracy in identifying parked cars. The experiment was run under ten different conditions, with YOLOv3 reaching the highest degree of precision with a probability of 90% and a precision of 96.88%. Contrarily, the accuracy of the Haar Cascade Classifier is 63.34% (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

As the number of automobiles on the road rises at an alarming rate, parking lots become more and more out of balance with cars. There are typically not enough parking spaces available when new buildings or public spaces are constructed. This makes cars stop and park on the side of the road, making it harder for them to find a parking space (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

To avoid causing traffic and wasting time looking for parking spaces, it is imperative that drivers and parking staff, managers, and the public are aware of the availability of roadside parking spaces. In this project, a system called "Smart Parking" will be created, specifically for street parking, to make it easier for drivers to see what parking spots are available and where the empty ones are (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

The haar cascade classifier approach is utilised in this study to identify automobiles, especially those that are idling on the side of the road. Experiments are carried out to identify the items approach after numerous procedures utilising the haar cascade classifier, and it eventually produces an output of a recognised object in a picture. A flowchart of the steps a photograph must take to get the results of vehicle object recognition is shown in ***Figure 6***.

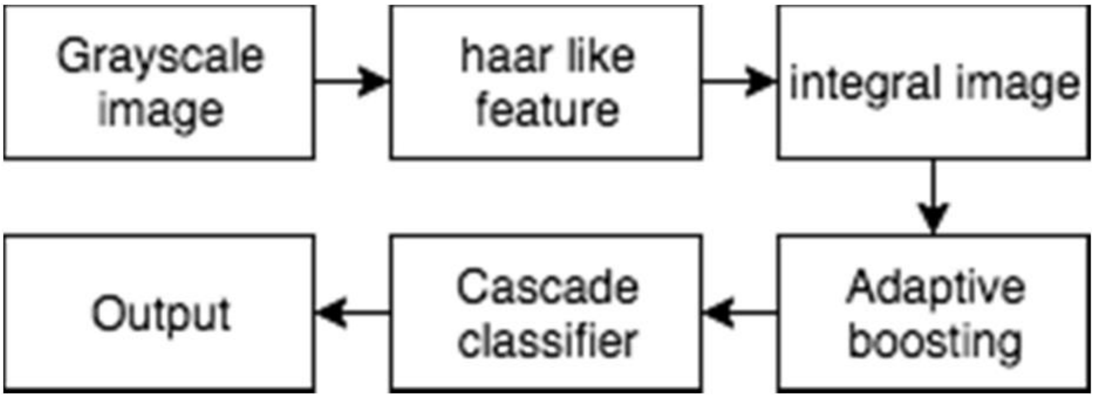


Figure 6: Car detection scheme uses the haar cascade classifier

Yolo (You Only Look Once) is a real-time item detection technique. A localizer or classifier that has been modified is used by the detection system to produce a detection. A model is included in a picture in many shapes and settings. The region in the image with the highest score will be used to decide the detection (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

A synthetic neural network called YOLO can identify objects in photos (ANN). This method divides the image into many sections and predicts what will occur adjacent to the bounding box and the likelihood of each area. The bounding boxes were then compared to the anticipated probability. YOLO provides various advantages over systems that use classifiers, including the capacity to watch the complete image as the test is being run and generate predictions based on the image as a whole (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

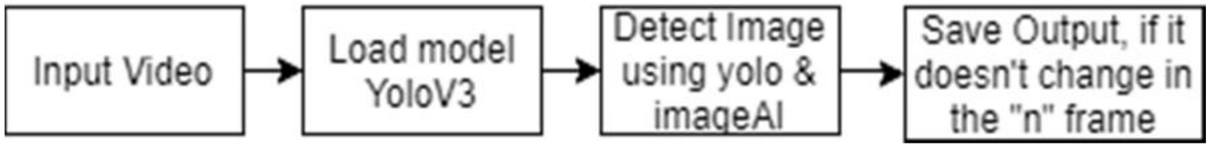


Figure 7: Flow detection of cars using YOLOv3.

Data for this experiment was gathered from a CCTV camera that was positioned at a height of 2.5 metres using the relative accuracy of the algorithms YOLOv3 and Haar Cascade Classifier. Ten alternative scenarios were used in the trial. The YOLOv3 algorithm has an accuracy of up to 96.88%, whereas the Haar Cascade Classifier has an accuracy of just 63.34% (D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, 2020).

2.2.4 Smart Parking-based System for smarter cities (S. Kazi, S. Khan, U. Ansari and D. Mane, 2018).

Public transportation is becoming outnumbered by private vehicles as the world grows more and more motorised. The demand for parking spaces grows along with the number of individuals who own cars. However, there aren't enough parking spaces accessible right now, and individuals don't know where they can park legally in their neighbourhood. Because of this, there are too many automobiles on the road, which makes it difficult for people to stroll down the street (S. Kazi, S. Khan, U. Ansari and D. Mane, 2018).

In order to address these issues, we propose a solution that would help individuals discover parking places online: a multilingual Android application. Using a convenience sampling approach, we show that this system will allocate a parking place based on the user's current location and the parking slot they pick. Finding the right spot is simple. Digital purchases are available, as well as those made through vending machines. To assist the system, detect the location and display the closest parking place and parking slot, as well as whether it is available, the end-user may create an account and log in with it. If not, the user will be sent to the following closest open slot (S. Kazi, S. Khan, U. Ansari and D. Mane, 2018).

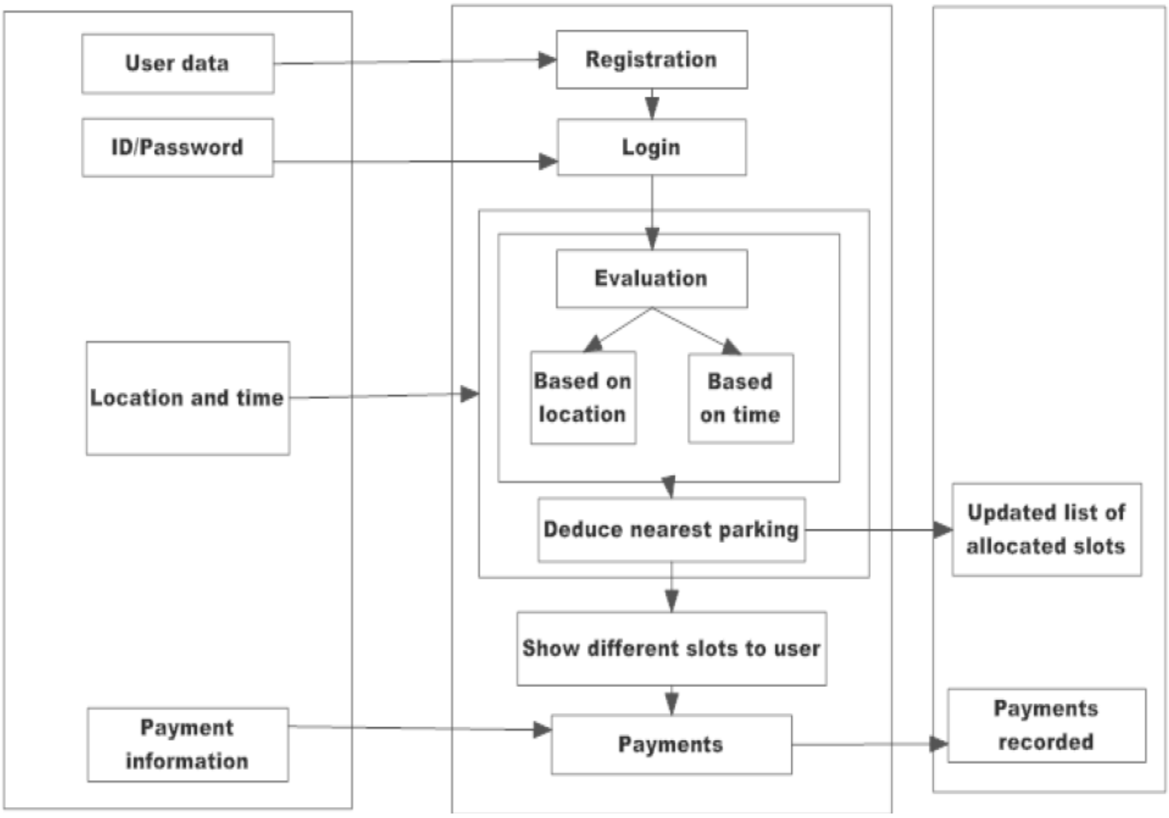


Figure 8: Block Diagram System based on smart parking for smarter cities

This suggestion is meant to deal with the ongoing parking problem that metropolitan areas experience. It uses several search algorithms to provide the user with the closest open parking place. It allows for dynamic real-time slot searching and navigation. We can do this by utilising techniques like max-flow algorithms or genetic algorithms (S. Kazi, S. Khan, U. Ansari and D. Mane, 2018).

The goal of this project is to provide users with a flexible and easy way to park their cars in locations that are both convenient and legal. If the suggested system is put into place, it will be able to overcome the shortcomings of the present traditional and application-based methodologies. Ineffective parking lot management is a major issue worldwide. Our suggested method will be able to somewhat solve the shortcomings of the existing system (S. Kazi, S. Khan, U. Ansari and D. Mane, 2018).

2.2.5 Towards a Smart Parking Management System for Smart Cities (P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

A range of parking-related issues may be resolved with the help of the Smart Parking Management System (SPMS). With the help of a smartphone app and sensor technology, this SPMS enables drivers to interact in real-time with parking infrastructure. A small-scale testbed was constructed as a functional prototype and put to the test in five scenarios (P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

The newest trend, known as "smart cities," aims to efficiently and intelligently automate the monitoring, access, and usage of infrastructure that supports vital services provided to citizens. The most crucial enablers of effective and secure administration of municipal resources, which are frequently few, are cutting-edge wireless sensing technology, machine learning strategies, 5G networks, as well as big data analytics software. These resources have the potential to greatly enhance people's lives if they are used intelligently, leading to higher quality education, more inexpensive healthcare, and more accessible, pleasurable, and environmentally friendly transportation (P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

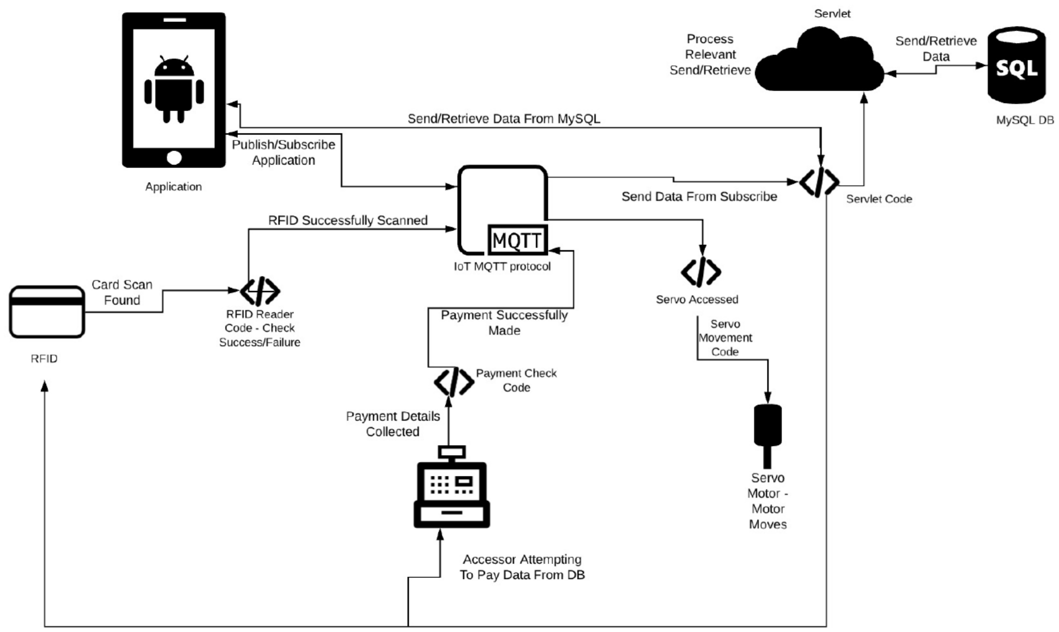


Figure 9: System Design Smart Parking Management System for Smart Cities

(P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

Figure 9 depicts the expected outcome of the system proposal (SPMS) as well as interconnections between its various components. In this approach, A servlet is linked to a SQL database, allowing data to be retrieved and transferred. Data is given and utilised by the Message Queuing Telemetry Transport (MQTT) protocol to the application/controller, which is controlled by code. The MQTT protocol, which collects, and analyses data as required, is a critical component of our system. When the criteria are met, this data may be used to regulate the servo motors that open as well as close the car park barrier (P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

The proposed an innovative Smart Parking Management System (SPMS) that provides drivers with factual facts about the availability of parking spaces and assists them in easily locating their parked vehicles, particularly in massive parking spaces, to alleviate the worry and tension that drivers endure when looking for a parking place and to shorten their search time. A small-scale testbed was utilised to evaluate the effectiveness of our SPMS, and a variety of situations were investigated. If future access to car parks in the actual world is granted, our testbed may be enlarged to carry out a trial run with actual vehicles and evaluate its efficacy in real-life circumstances (P. Melnyk, S. Djahel and F. Nait-Abdesselam, 2019).

2.2.6 Towards a Smart Parking Management System for Smart Cities (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

One of the most difficult problems in crowded cities is finding adequate parking places. India has one of the lowest ratios of vehicles to parking spaces worldwide. Only 1800 parking spaces are available in our nation for every 25,000,00 registered cars. To assist drivers with parking space scheduling, our article describes an intelligent automotive parking system that makes use of cloud based IoT technologies. A unique QR code is given to each customer who rents a parking spot. It is now feasible to monitor and authenticate vehicles without direct interaction (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

In-ground sensors alert the parking police when a second car is parked in a certain location. As a result, we may make sure that our cars are parked at a location that is handy for us (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

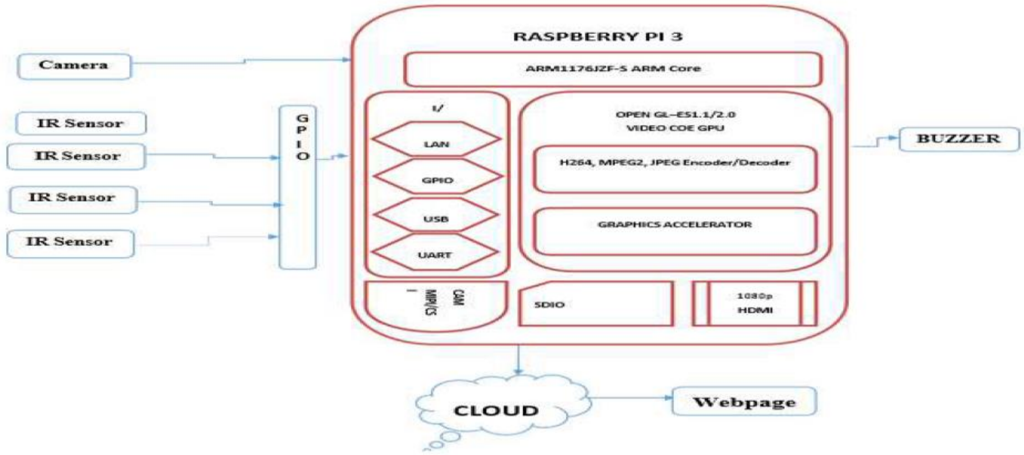


Figure 10: Block Diagram Intelligent Car Parking System Using Raspberry PI

(B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

Parking spaces in the system are equipped with infrared sensors and a camera run by a Raspberry Pi. Infrared sensors are used to determine if a car is there or not, and the available parking spaces are shown. On the parking destination site, the user can register any open, vacant spots. The user gets a QR code right away (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

Given the rising number of vehicles on the road, a better parking system is necessary. Auto parking has historically been unorganised and done at random. To address each of these issues, a wider implementation of our suggested approach is required. This initiative provides substantial benefits over the present strategy. Benefits of the suggested plan include less impact on the environment, practical methods, accurate monitoring, and more (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

The abundance of parking spaces would help to lessen traffic congestion brought on by tired drivers looking for a parking space. These ingenious parking spaces will bring in new revenue. Total automation of the process will also do away with the necessity for pointless physical labour. Additionally, these clever parking lots aid in limiting resource depletion (B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, 2021).

2.2.7 IoT-Basedsed Smart Parking System for Large Parking Lot (M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, 2021).

Since there aren't enough parking places in metropolitan areas, car parking has long been a major issue. In this study, we provide a smart parking system for large parking lots based on the Internet of Things that may be used to effectively manage the parking system by providing information on the next available parking slot via a mobile application, therefore reducing parking-seeking congestion. To effectively manage the parking system, a successful cloud-based smart parking system solution built on the Internet of Things technology has been designed to direct consumers to the closest available parking place (M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, 2021).

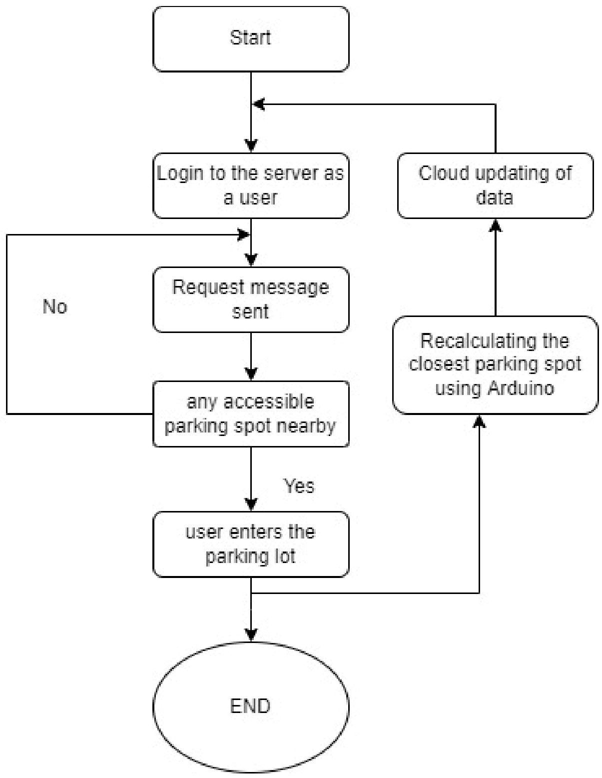


Figure 11: Flow Large Parking Lot Smart Parking System Powered by IoT.

The system's entire flow is seen in ***Figure 11***. All information about the availability of parking spaces is stored on an open IoT platform, and the motorist will have access to the system through an app made with the BLYNK mobile app. A parking place request message will be sent by the user after connecting to the server. The user will be permitted to enter the car parking area and directed to the nearest accessible place if a parking space becomes available (M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, 2021).

By informing users of the next open parking space, the built-in IoT Smart Parking System may effectively regulate the parking system and ease parking-seeking congestion. This parking system might be employed in colleges, airports, and retail centres to lessen the number of people who need to find parking and save them time (M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, 2021).

2.2.8 Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University (N. Sieck, C. Calpin and M. Almalag, 2020).

Campuses of colleges frequently have parking issues. Smart technology-using organisations do not have a common language. Examining a proof-of-concept design for putting in a smart parking system in a parking lot at a medium-sized institution is the goal of this study. The results show that the recommended strategy has some intriguing potential (N. Sieck, C. Calpin and M. Almalag, 2020).

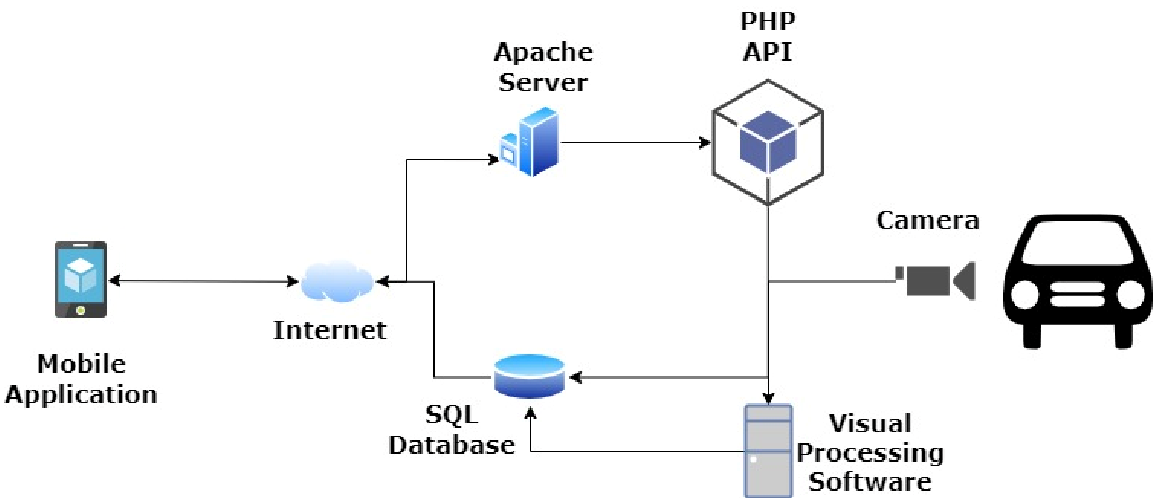


Figure 12: System Architecture Internet of Things (IoT)-Based Machine Vision Smart Parking at a Smart University.

Figure 12 depicts the system that we have designed. To find available parking spaces, the system starts with cameras gazing over parking lots. This data is input into a database, which the programme accesses using a PHP API. We are aware of the potential risks associated with this solution, including wind and weather exposure (N. Sieck, C. Calpin and M. Almalag, 2020).

As the user would access the app while driving, they would like to develop a hands-free method of using the app. The user of the Siri Shortcut function might utter a personalised phrase to be guided to the closest parking lot with free space. Only reserved parking spaces could be reserved using the app, and it would only work if the user was physically within a mile of the school (N. Sieck, C. Calpin and M. Almalag, 2020).

2.3 Comparison Table with Exiting Parking Systems

Ref	Problem	Contribution	Approach	Explanation of tools used	Limitations
[1]	If parking is difficult to find, inadequate, inconvenient, or pricey, visitors will be unsatisfied and may depart. Therefore, a lack of parking places can cause issues for both residents and cars.	Helps to solve the problem of urban transit parking in smart cities by building a system that is open and available to the public.	Examines the possibility of creating a mobile application that makes use of the Internet of Things.	The platform that uses the cloud to build web applications.	There is just a little database that this program supports.
[2]	Parking can be difficult to find for many drivers, which increases fuel use, costs, and emissions.	Numerous studies have concentrated on the distribution, accessibility, and management of parking in the city to alleviate the parking problem.	Image analysis, picture segmentation, image enhancement, noise reduction, geometric transformations, and image registration are all part of the Image Processing Toolbox.	The quadtree technique was used to alter and identify images in MATLAB.	It may improve the accuracy of the parking system to evaluate the detection rate using a variety of facilities, such as large parking lots, indoor parking, parallel parking, and high-quality cameras.
[3]	The rapid increase of vehicles leads to	Recognising objects in a photo by using an	To identify vacant spots and parked	To find parked cars, the YOLOv3	If the App's user base grows, a

	an imbalance between vehicles and parking spaces.	artificial neural network (ANN).	automobiles, a camera is positioned on the side of the road that doubles as a parking lot.	detection and the haar cascade classifier were utilised.	community -based service may be added, enabling people to find appropriate parking spots even farther than the roadside.
[4]	People prefer to leave their cars in the middle of the road, which endangers pedestrians and makes it more likely that their cars will be towed for unlawful parking.	Wireless Sensor Nodes were the employed technology (WSN). Therefore, if a user makes a request for a parking or free spot, the request will be sent to the closest nodes using WSN technology.	The vacant parking spaces were found using the min-max method.	Developed a bilingual Android application to help consumers locate parking spots online.	Being a machine, the system will undoubtedly malfunction at some point. In another scenario, it can malfunction and lead to improper car parking.
[5]	Parking delays and unnecessary tension make the traffic situation even worse.	To reduce drivers' time spent looking for an open parking space in big multi-story car parks and to make it easier for them to locate their parked vehicles, SPMS offers a sensing platform backed by a mobile app.	Intelligent Transportation Systems (ITS) seek to drastically alter the transportation industry to successfully tackle the disastrous effects of escalating traffic congestion.	On a compact testing bench, Phidget hardware was used to evaluate the proposed SPMS.	The testbed might be expanded to undertake a pilot test with actual vehicles and assess its efficacy in actual-world scenarios.
[6]	It's challenging for drivers to locate a good parking space. Another major issue is overcrowding of parking spaces and illegal parking.	An intelligent IoT-based cloud-based parking system that aids drivers in scheduling parking spaces.	The hardware of the gadget includes a Raspberry Pi 3 computer, IR sensors, buzzers, and a camera for reading QR codes.	Programming language Python. The user may locate parking places for their car using the Open CV approach.	The battery life of sensors is gradually getting shorter due to the increasing accuracy requirements.
[7]	People hunting for a parking	To find parking spaces, a smart	Uses an IoT component	A smart parking	By using the

	place during holidays and peak hours contribute to parking shortages, high parking costs, and traffic congestion.	car scanning the proposed system uses QR codes. The information about the parking lot, floor, and parking space ID is obtained by using a smartphone to decode the QR code.	on-site to manage available parking spaces and offer proof in the form of a webpage for bookings of parking spaces.	system that combines IEEE 802.15.4 Wireless Sensor Network (WSN) and Ultra-High Frequency (UHF) Radio Frequency Identification (RFID) technologies is used to guide automobiles to the nearest accessible area with the aid of a built software application.	instructions in the developed mobile application, parking seekers might locate the nearest parking space by connecting it to a navigation system. In addition, the right algorithms may be created to allow parking seekers to book their slots in advance.
[8]	The average motorist wastes seventeen hours per year looking for parking, which costs an estimated $72.7 billion per year in gasoline and pollution, or $345 per driver each year.	The authors used a proof-of-concept framework to construct a smart parking system in a midsize university parking lot.	The system technique comprises image processing, vision-based technologies, and license plate identification employing cameras to analyze parking spot availability in a parking lot.	The system makes use of tools such as the Apache Server, a mobile application, the PHP API, and Visual Processing Software.	The application developed can be hands-free and the parking slots can be reserved in advance.

2.4 Advantages of Smart Parking

- **Improved parking** - The best location is found by users, saving them time, money, and effort. The parking lot soon fills up, enabling companies to effectively utilise the space that is available.
- **Lessen Traffic** - Fewer automobiles are needed to travel around looking for parking spaces, which improves traffic flow.

- **Less pollution** - Every day, one million barrels of oil are burned while people look for parking. The amount of time spent travelling will be greatly reduced by the ideal parking solution, which will reduce daily car emissions and eventually the environmental impact on the world.
- **Data in Real Time and Trend Analysis** - Over time, a smart parking system can provide data that demonstrates connections and trends between parking lots and their patrons. If lot owners want to improve drivers and make changes, these patterns may be very helpful.
- **Superior User Experience** - A smart parking system will condense the whole user experience into a single task. The payment of the driver, location searches, spot identification, and time alerts are all steps in the process of getting to the destination.

2.5 Disadvantages of Smart Parking

- **The high cost of construction or installation** - A dependable, effective parking management system can be rather expensive. This is a result of the system's complicated structure. Due to components like the statistics feature, automatic ticketing, statistical reports, and countless others, everything is expensive. Some organisations might not be equipped with the necessary tools.
- The system still requires periodic maintenance from the organisation despite being automated. This is done to ensure that everything is working well and without any mistakes. The maintenance might be done once a month.
- **Operation** - The parking control system is strange to a lot of people. As a result, they could find it difficult to use, which would make parking more complicated.
- **Breakdown** - Given that a system is a machine, it might malfunction at some point. When this occurs, cars may not be able to access structures or move vehicles that are parked within. Additionally, it could malfunction and lead to improper parking of automobiles.

Chapter 3 – Design Methodology & Implementation

3.1 Introduction

Our goal is to create a smart parking system that would help users locate open parking spaces in congested regions. To determine if a parking spot in the obtained footage is occupied or not, edge detection and coordinate bound pixel sections are used in combination by the OpenCV library and Python for digital image processing. From picture acquisition to image detection, ***Figure 13*** uses image processing methods from Python and the OpenCV library. In this section, the proposed methodology is explained with a diagram in detail. Implementation with steps is explained in detail.

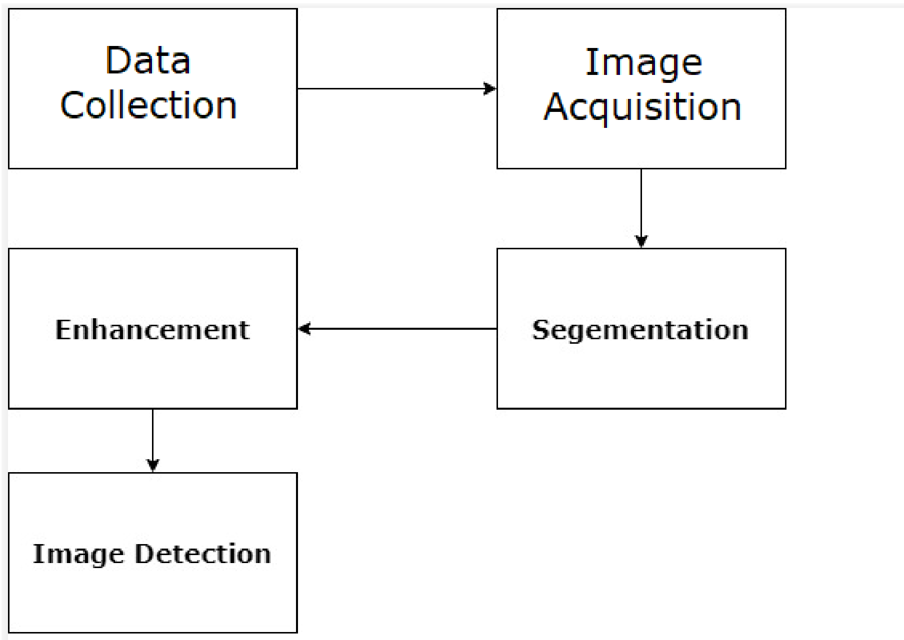


Figure 13: Schematic of the system module in block form

3.2 Smart Parking Prototype

This study looks at the process for developing a smart parking technology prototype. Two toy automobiles are used to show off the prototype. The parking place is composed of black cardboard and is designated with white tape. At the entry and departure locations, two mobile IP cameras are being utilised to record the licence plate information of the vehicles. At the entry and departure locations, a mobile camera is set at a certain angle to record the vehicle's licence plate. At the entry point, the car's number plate is photographed and added to the database for records. At the departure point, the car number plate is scanned to indicate that the vehicle has left the area and its number is removed from the database. ***Figure 14*** demonstrates how the smart parking system operates.

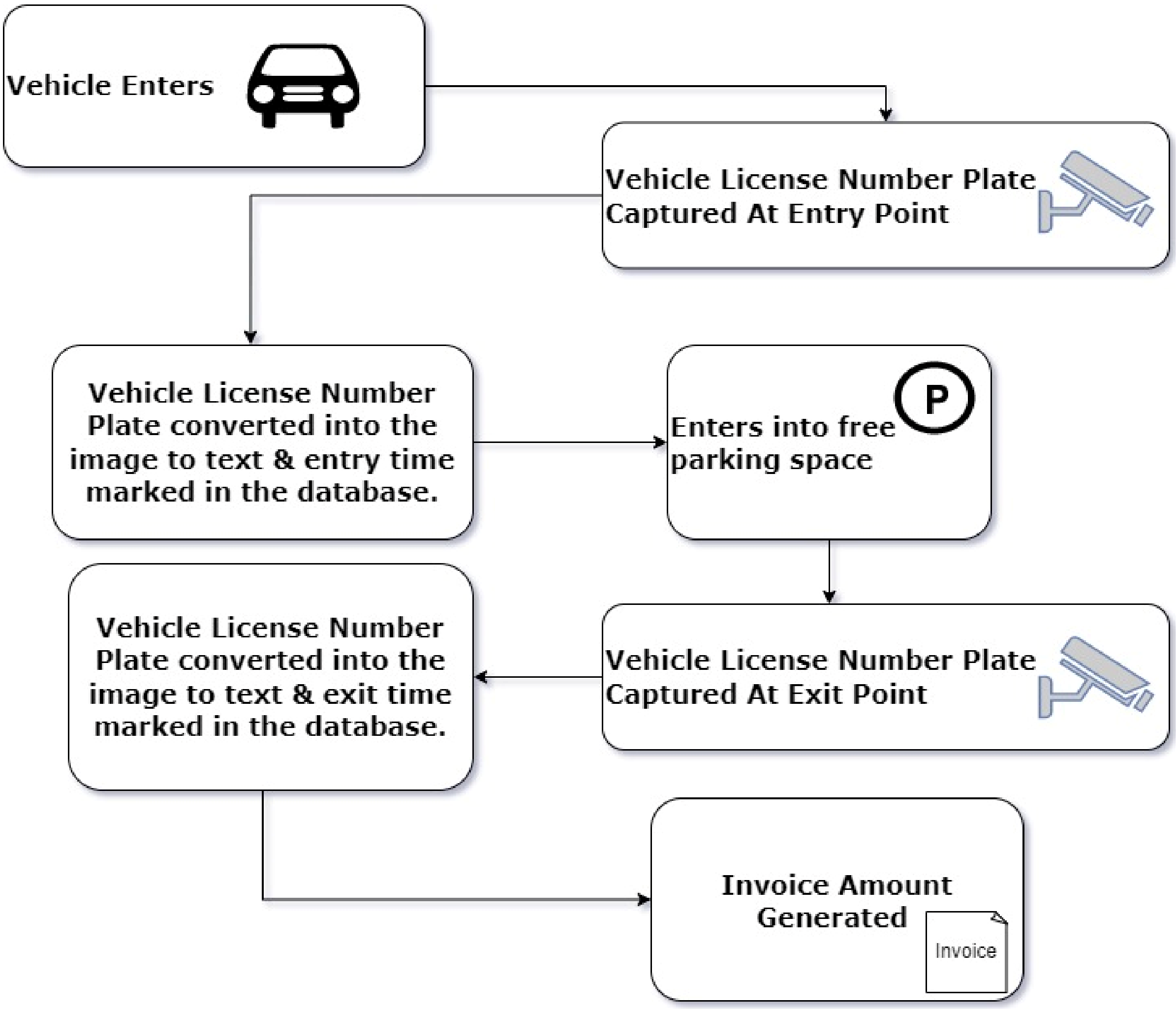


Figure 14: Workflow of The Design System

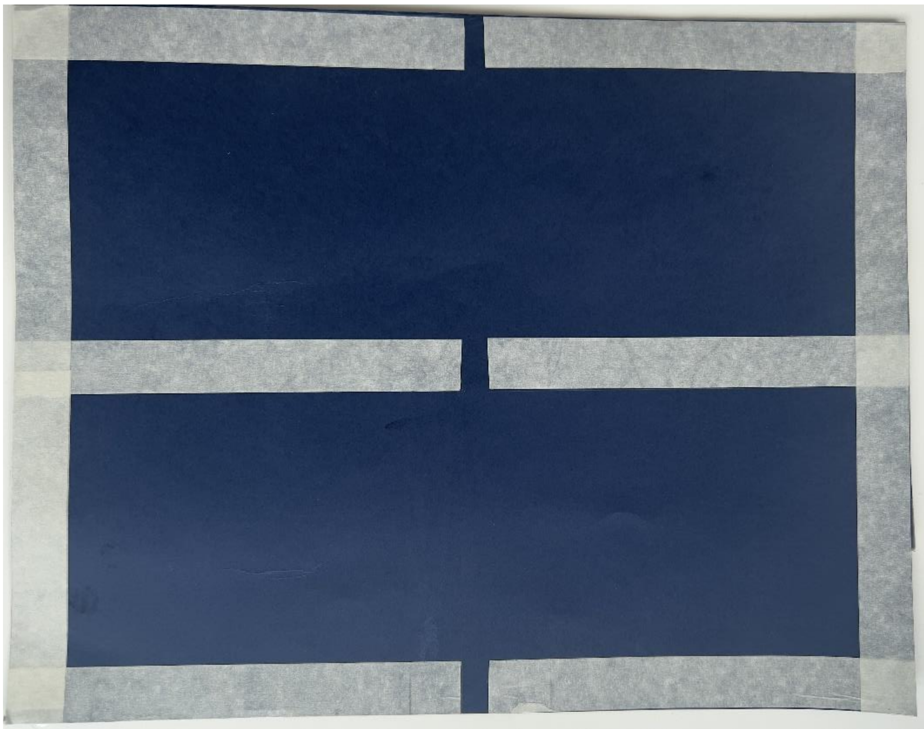


Figure 15: Prototype Parking Lot (Top View)

The Prototype Parking Lot's upper perspective is seen in ***Figure 15***. In the demo prototype, there are a total of four parking spaces.

Figure 16: Two toy cars are used in the smart parking prototype.

Two model automobiles utilised in the smart parking prototype are shown in ***Figure 16***. Demonstration licence plates, utilised in the smart parking prototype, are seen in ***Figure 17***. The project's demo licence number is generated at random.

Figure 17: Displaying Licence Plates.

Figure 18 shows the license number plates captured on the entry and exit points.

Figure 18: Licence Plates Captured at Entry & Exit Points.

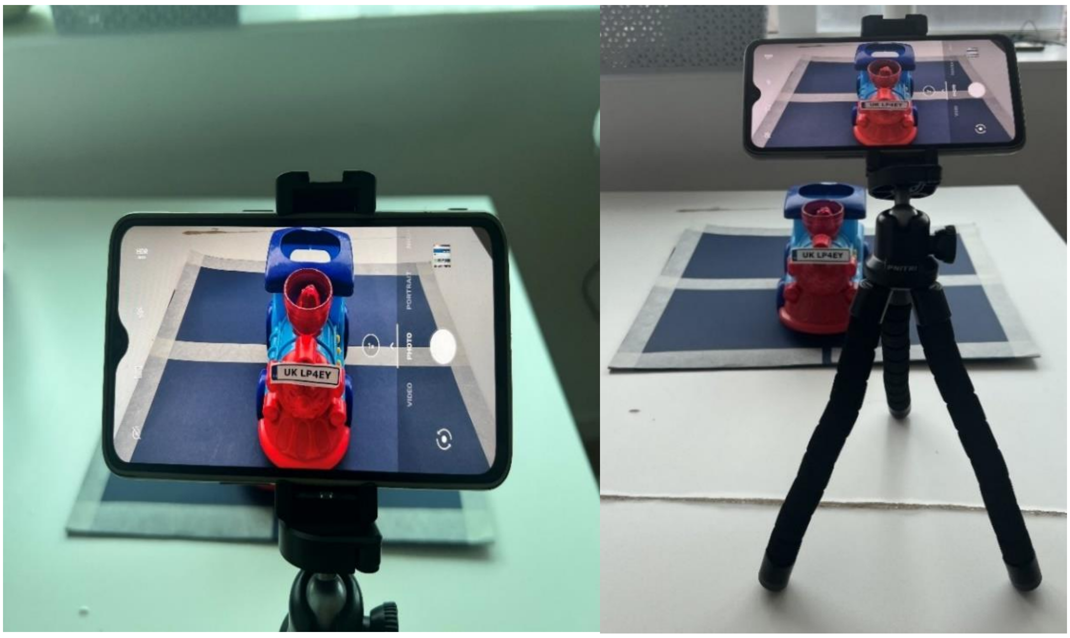


Figure 19: Camera Mount to Capture License Plate at Entry Point.

Figure 19 shows at the entrance, a camera mount will capture the licence plate. ***Figure 20*** shows at the exit point, a camera mount will capture a licence plate. Once the car enters, the data capture is recorded in the database at the exit point, the licence plate is scanned to delete the entry from the database and signal the vehicle's exit. The automobiles in the parking lot are seen in ***Figure 21***.

Figure 20: License plate capture using a camera mount at an exit.

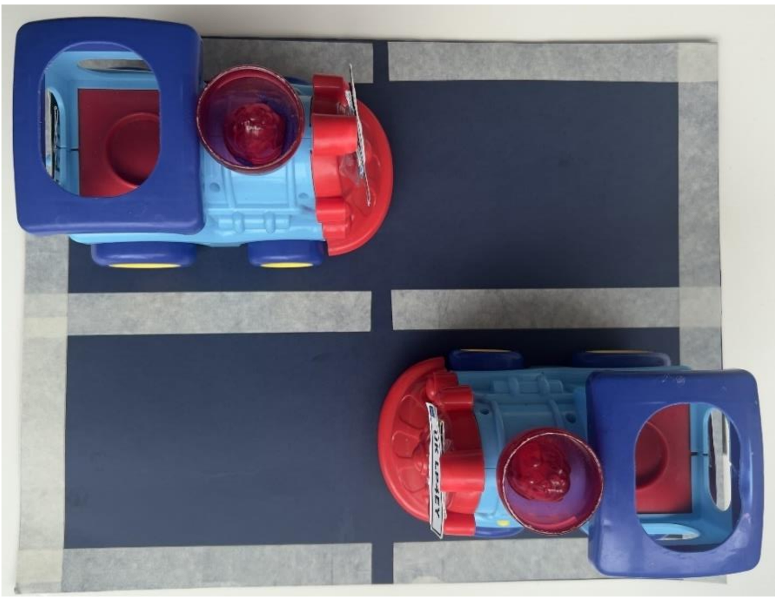


Figure 21: Vehicles Parked in The Parking Lot

3.3 Software Deployment

3.3.1 Programming Environment

The Python 3 environment serves as the foundation for all the project's programming ideas. Python is a high-level, all-purpose programming language that is interpreted. Using significant indentation, its design philosophy places a strong premium on code readability. Its language elements and object-oriented methodology aim to help programmers create logical, straightforward code for both small- and large-scale projects. (Kuhlman Dave, 2012). Python is a more rational programming language. Python is thus an excellent programming environment.

3.3.2 Chosen Library

Python libraries are essential for creating machine learning, data science, computer vision, picture, and data transformation applications, among other things (Advani, V, 2020). Before beginning to programme, the necessary libraries should be imported.

Most of the libraries used in the progress of programming are offered in this section:

1) NumPy

A Python package called NumPy makes it possible to use enormous, multidimensional arrays and matrices as well as a great variety of high-level arithmetic operations on these arrays (Anon, n.d).

2) Cv2 Python Library

To solve computer vision difficulties, a group of Python bindings named OpenCV-Python was developed. For Example, The CV2 function imread () loads an image from the specified file. If the image cannot be read, this method returns an empty matrix (owing to an unsupported or incorrect format, a missing file, inadequate permissions, etc.).

3) Pickle Python Library

Pickle is often used to serialise and deserialize a Python object structure. It entails converting a Python object into a byte stream to transfer data over the network, save it in a file or database, or maintain programme state consistency across sessions.

4) CV Zone

The computer vision package CVzone makes it simple to do tasks like face identification, hand tracking, position estimation, and other AI operations. It also supports image processing. It primarily makes use of the OpenCV and MediaPipe packages.

5) Easy OCR

EasyOCR is a Python-based library for employing a ready-to-use OCR model. You won't have to bother about pre-processing or modelling using this package. With just a few lines of code, you can improve the accuracy of the OCR.

6) Imulits

With OpenCV and both Python 2.7 and Python 3, several useful methods to make common image processing operations like translation, rotation, scaling, skeletonization, and presenting Matplotlib pictures simpler. This module is utilized for identifying the vehicle's license plate number.

7) Sqlite3

SQLite is a C library that provides a lightweight disk-based database that does not require a dedicated server operation and may be accessed using a slightly different variation of the SQL query language. SQLite can be used by some apps to store actual information.

8) Date Time

Classes for manipulating dates and times are available in the Date Time module. The current entrance time and the exit time when the car enters and quits the parking area are displayed in this project by importing this module. The same data is presented on the web application and helps to produce an invoice for the user.

3.4 Implementation

1) Video Captured from Live Camera

Live footage of the prototype parking lot is recorded using the Python script camstrime.py. To record live video, we utilise an Android mobile device with an IP webcam. The same is seen in ***Figure 22***.

Figure 22: Android smartphone with an IP camera

```python
# capture live video strime
URL = "http://192.168.0.180:8080/video"
capture = cv2.VideoCapture(URL)
```

Figure 23: URL to capture live video stream

Figure 23 displays the URL for the prototype camera's live video feed. The while loop code for reading, displaying, and stopping a video stream is shown in ***Figure 24***.

```python
while 1:

    # read video stream
    _, frame = capture.read()

    # show the video stream
    cv2.imshow('LiveStrimeScreen', frame)


    # to stop the stream
    if cv2.waitKey(1) == ord("q"):
        break

# close the resorces
capture.release()
cv2.destroyAllWindows()
```

Figure 24: While loop logic for a live video stream

2) Video To Frames

The live video stream recorded by the camera is converted into frames via the video to frame.py file. ***Figure 25*** displays the necessary Python code and CV2 library.

```python
00_video_to_frame.py > ...
import cv2
# import os
# import time
#working

path = r'data\02_Parking_Lot.mov' # r'data\carPark1.MOV' # path1 = 'D:\test1'
vidcap = cv2.VideoCapture(path)

# address = "http://192.168.1.2:8080/video"
# vidcap.open(address)

count = 0

while (vidcap.isOpened()): #success:
  #time.sleep(3)
  success,image = vidcap.read()
  print('Read a new frame: ', success)
  cv2.imwrite("data/img/frame_%d.jpg" % count, image)
  #cv2.imwrite(os.path.join(path, "straight%d.jpg" % count), image)     # save frame as JPEG file
  #gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
  #cv2.imwrite(os.path.join(path1, "straightbw%d.jpg" % count), gray)  # save frame as JPEG file
  count += 1
```

Figure 25: Video to Frames Convert Code

3) Parking Space Picker

With the help of the parking_sapce_picker.py code we mark the total available parking space in a parking lot. We use one of the videos to frame converted image from our previous step. We use mouse click events to draw a rectangle over a parking space, left mouse click is used to mark a parking space & right mouse click to delete a wrong marked parking space, ***Figure 26*** shows total four marked parking spaces which creates a .pkl file in the system and compares the same with the live video footage to detect the vacant parking space. ***Figure 27*** shows the full code used to implement the parking space picker.

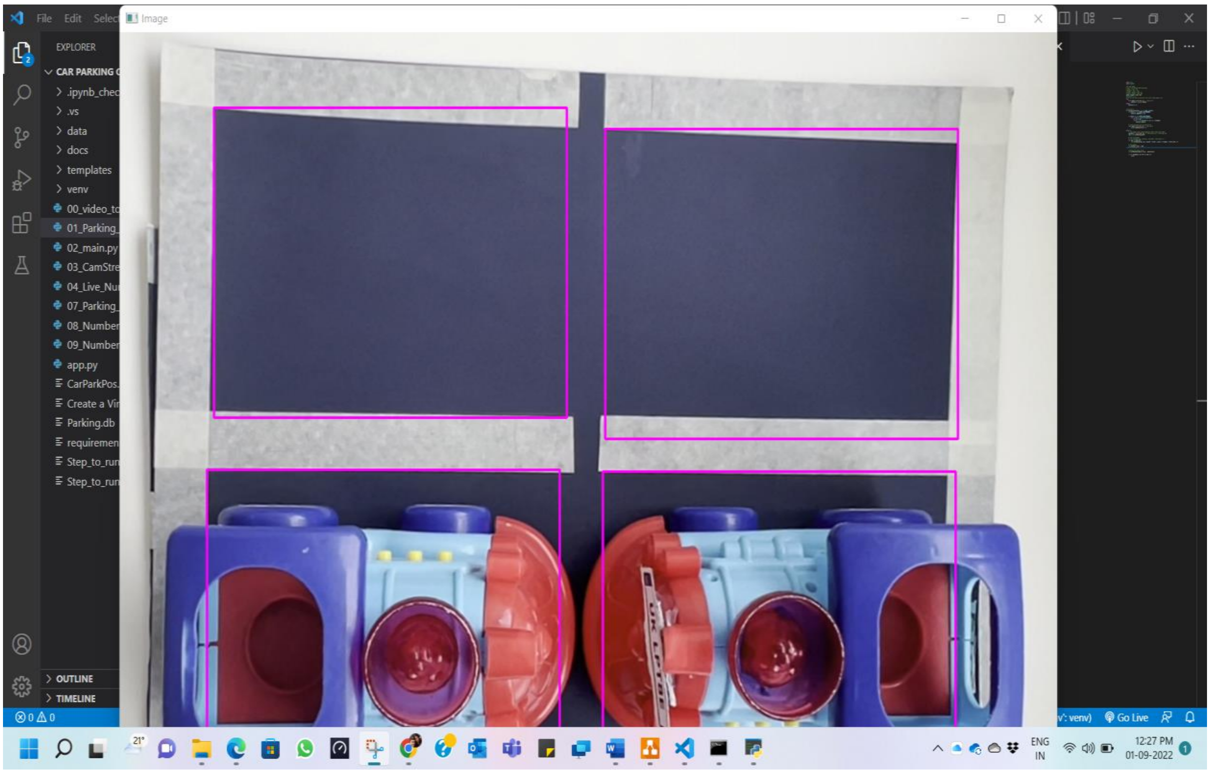


Figure 26: Video to Frames Convert Code

```
01_Parking_Space_Picker.py > mouseClick
import cv2
import pickle

# # load image
# img = cv2.imread('data/img.png')
# width = 156 - 50
# height = 240 - 193
# width, height = 106, 47
# width, height = 400, 300
width, height = 450, 350
# postList = []
# try to find the old position list else create empty list
try:
    with open('CarParkPos.pkl', 'rb') as f:
        postList = pickle.load(f)
except:
    postList = []


# function's
def mouseClick(events, x, y, flags, params):
    if events == cv2.EVENT_LBUTTONDOWN:
        postList.append((x, y))

    if events == cv2.EVENT_RBUTTONDOWN:
        for i, pos in enumerate(postList):
            x1, y1 = pos
            if x1 < x < x1+width and y1 < y < y+height:
                postList.pop(i)

    # write position list to pickle file
    with open('CarParkPos.pkl', 'wb') as f:
        pickle.dump(postList, f)

while 1:
    # load image (load image dynamicaly every time like video)
    img_path = 'data/img3.jpg' # 'data/img1.jpg' # 'data/img.png'
    img = cv2.imread(img_path)

    # draw rectrangle
    # cv2.rectangle(img, (50,193), (156,240), (255,0,255), 2)
    for pos in postList:
        cv2.rectangle(img, pos, (pos[0] + width , pos[1] + height), (255,0,255), 2)

    # show image
    cv2.imshow('Image', img)
    # cv2.waitKey(1)

    # detect the mouse click
    cv2.setMouseCallback('Image', mouseClick)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Figure 27: Parking space Picker Code

4) Parking Count with Database

This is the prototype's primary logic code file. The parking count with db.py file's primary function is to relay live footage from the installed cameras used to inspect the parking spot. Digital image processing is carried out in this section of the code, where coloured images are converted into grayscale images, image blur, convert the image into binary, remove noise from the image, and make pixel values thicker and connect with the local database to update the parking lots that are currently available.

As a result, the live video stream is compared to the *.pkl* file generated in the preceding step, digital image processing is performed, occupied parking spaces are marked in red, and vacant spaces are marked in green. The total number of available spaces is then calculated, and the local database is updated.

```python
# check the frame count to reset the frame
if cam.get(cv2.CAP_PROP_POS_FRAMES) == (cam.get(cv2.CAP_PROP_FRAME_COUNT) - 5):
    cam.set(cv2.CAP_PROP_POS_FRAMES,0)

# read frame from video
success, img = cam.read()

# convert to grayscale
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# add image blure
imgBlure = cv2.GaussianBlur(imgGray, (3,3), 1) # play with the parameter here (3,3), 1

# convert to binary image
imgThreshold = cv2.adaptiveThreshold(imgBlure, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 25, 16) # play with the parameter here (25, 16)

# remove the dot's noice between ()
imgMedian = cv2.medianBlur(imgThreshold, 5) # play with the parameter here 5

# make pixel values more thicker
kernel = np.ones((3,3), np.uint8)
imgDilate = cv2.dilate(imgMedian, kernel, iterations=1) # play with the parameter here
```

Figure 28: Digital Image Conversion Code

5) Scan the Vehicle License Number Plate

At the entry and exit locations, this section of the code scans the licence plate of the vehicle. With the aid of the algorithm, the scanned images are transformed from image to text with canny edge detection. Additionally, the local database's entrance and departure points are where the data is updated. ***Figure 29*** Additionally essential to the translation of images to text is the Python module *easyocr*.

```python
while 1:

    # check the frame count to reset the frame
    if capture.get(cv2.CAP_PROP_POS_FRAMES) == (capture.get(cv2.CAP_PROP_FRAME_COUNT) - 5):
        capture.set(cv2.CAP_PROP_POS_FRAMES,0)

    # read video strime
    _, frame = capture.read()
    # print(frame.shape)

    scale_percent = 50 # percent of original size
    width = int(frame.shape[1] * scale_percent / 100)
    height = int(frame.shape[0] * scale_percent / 100)
    dim = (width, height)

    # resize image
    frame = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)

    # print(frame.shape)
    # frame.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
    # frame.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
    # frame.set(cv2.CAP_PROP_FPS, 25)

    # Convert colored image into grayscale formate
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply Filter and Edge Detection
    bfilter = cv2.bilateralFilter(gray, 11, 17, 17) # Noise reduction
    edged = cv2.Canny(bfilter, 30, 200) # Edge detection

    # Find Contours and Apply Mask
    keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    contours = imutils.grab_contours(keypoints)
    contours = sorted(contours, key = cv2.contourArea,reverse=True)[:10]
```

Figure 29: License Number Plate Scanning Code

Chapter 4 – Results and Analysis

4.1 Introduction

The outcome of the digital image processing-created smart parking system prototype is shown in this chapter. displays a comparative analysis of the test results in tabular form. The results of the developed prototype, which is portable and completely functioning, are presented in this chapter.

4.2 Results

4.2.1 Image Frame Converted from Live Video Stream

Figure 30 shows the image frames that are converted from live video feed.

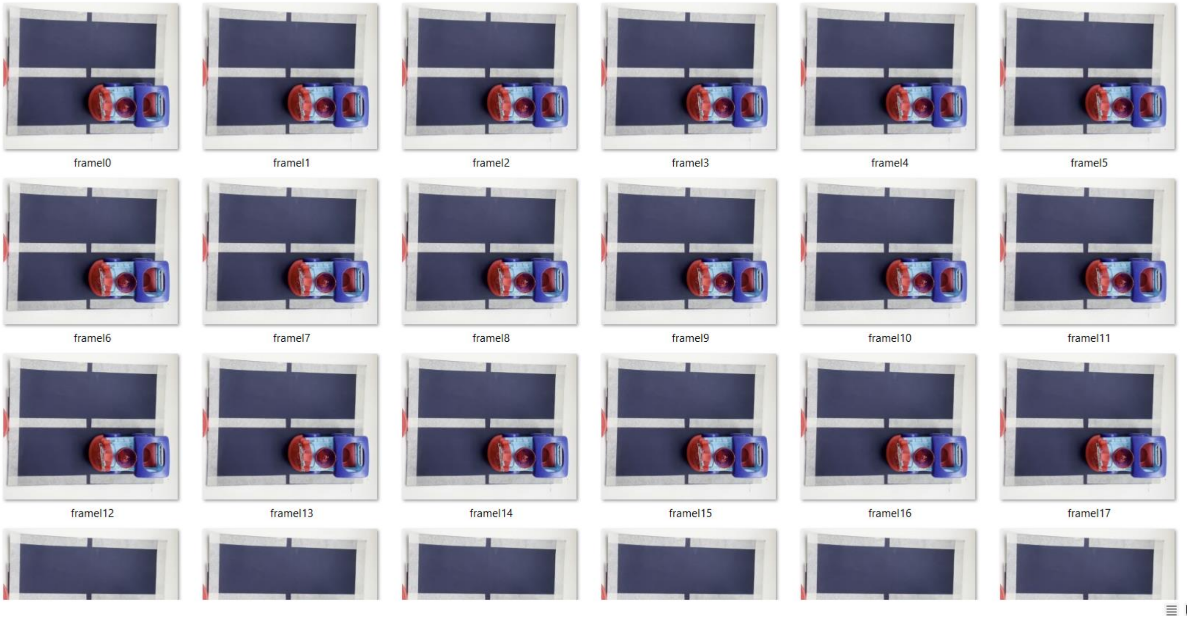


Figure 30: Converted Image Frames from Live Video Feed

4.2.2 Parking Space Picker

Figure 31 shows Four Parking spaces marked on the prototype created.

Figure 31: Parking Space Marked

4.2.3 Total Number of Parking Counts

Figure 32 demonstrates how the system can determine the right amount of parking from the total number of parking spaces. Digital image processing uses the logic that when the marked parking area's pixel count rises above 8000, the colour of the rectangle changes to red and updates the total number of parking spaces in the local database; conversely, when the pixel count drops below 8000, the colour of the rectangle changes to green and updates the local database.

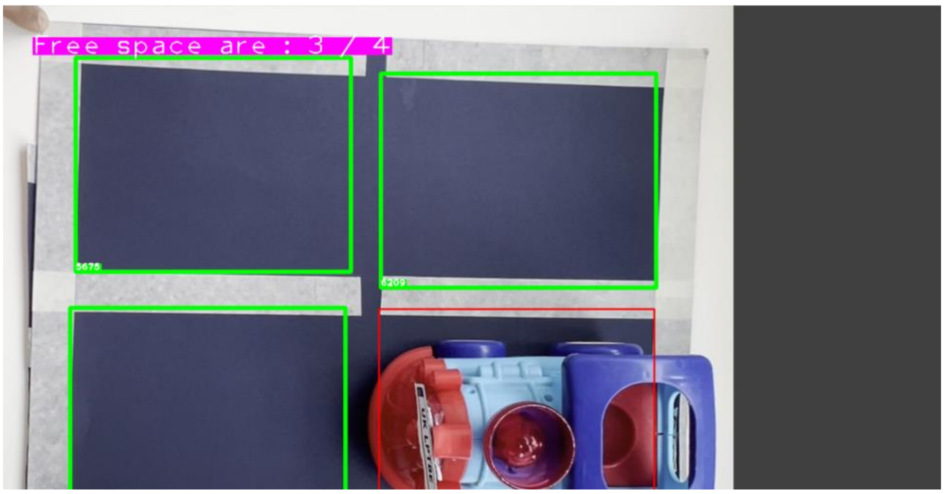


Figure 32: Total Number of Parking Spaces

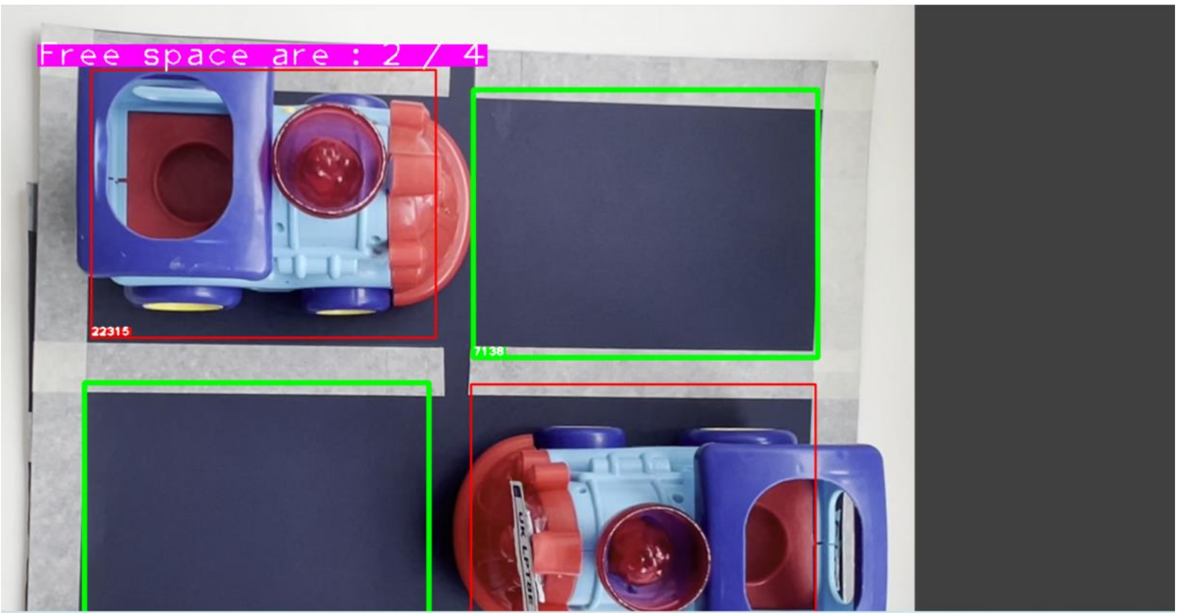


Figure 33: Pixel Count of the Parking Space

In ***Figure 32***, the top left parking lot has 5675 pixels, and the rectangle's colour is green. In ***Figure 33***, the top left parking lot has 22315 pixels, and its pixel count has been updated to reflect the revised number of available parking spaces.

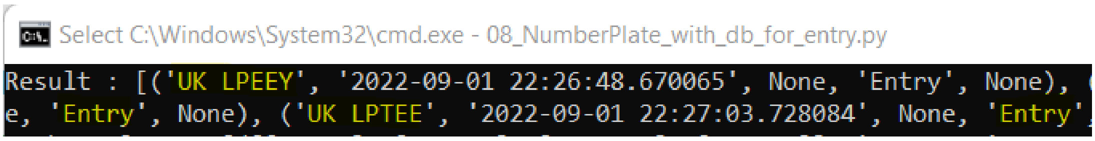
4.2.4 Number Plate Scanning at Entry & Exit Point

Figure 34 & ***Figure 35*** shows the two vehicles entry & the number plate scanned at the entry point.

Figure 34: Vehicle One Entry in the parking lot

Figure 35: Vehicle Two Entry in the parking lot

Figure 36 shows the entry marked in the database for two vehicles.



Select C:\Windows\System32\cmd.exe - 08_NumberPlate_with_db_for_entry.py

```
Result : [('UK LPEEY', '2022-09-01 22:26:48.670065', None, 'Entry', None),
e, 'Entry', None), ('UK LPTEE', '2022-09-01 22:27:03.728084', None, 'Entry'
```

Figure 36: Vehicle Entry in the Database

Figure 37 & ***Figure 38*** shows the two vehicles exit & the number plate scanned at the exit point.

Figure 37: Vehicle One Exit in the parking lot

Figure 38: Vehicle Two Exit in the parking lot.

Figure 39 & ***Figure 40*** shows the Exit marked in the database with the exit time for two vehicles and generates the invoice for the vehicles.

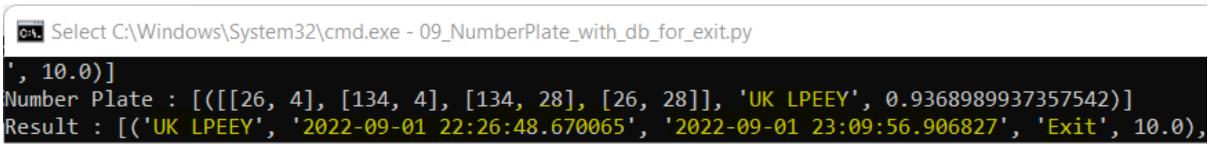


Figure 39: Vehicle One Exit in the Database

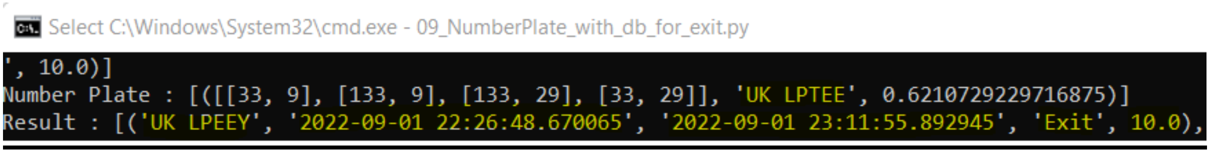


Figure 40: Vehicle Two Exit in the Database

4.2.5 Results Displayed on the Web Portal

Figure 41 depicts the outcomes shown by the online application. The outcome is shown through the web application. Parking area, available parking, total parking spaces, the vehicle's licence plate number, the entry and exit times, the condition of the car, and the parking fee.

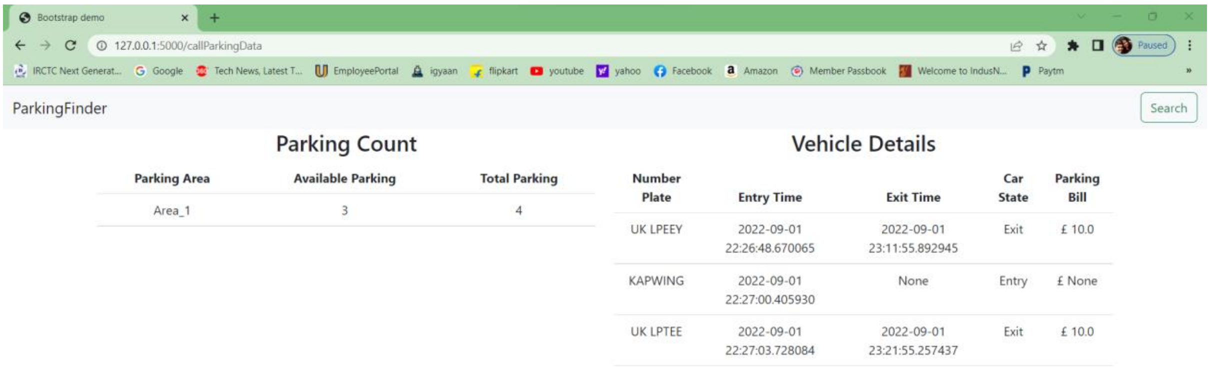


Figure 41: Outcomes Shown by The Online Web Application

4.2.6 Analysing the Portability of a Prototype in a Real Parking Lot

To test our prototype system, we captured a video of the parking lot using a drone camera. ***Figure 42*** demonstrates how well our prototype system is performing and provides the entire number of available slots. To evaluate the portability of the prototype system, the same implementation steps from ***Chapter 3*** were used. The total number of free parking spots is shown in ***Figure 42***. All the parking spaces that are occupied in ***Figure 42*** are over 900 pixels because of digital picture processing.

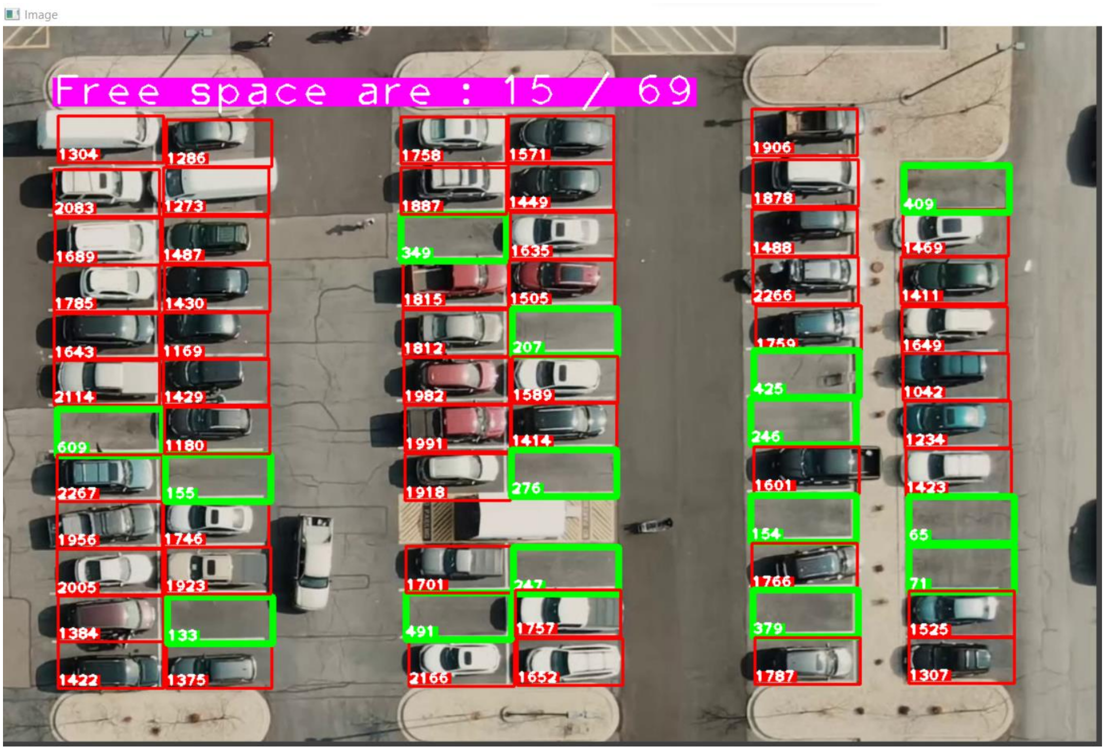


Figure 42: Outcomes Analysing the Portability of a Prototype System

Chapter 5 – Conclusion and Future Work

Cities that wish to become more efficient and have a lot of traffic issues should invest in smart parking. This may also be intriguing in terms of commercial real estate. The system can assist in lowering pollutants, the amount of time spent looking for parking spaces, and expenditures, particularly for big parking lots in front of shopping centres, businesses, and industrial buildings.

In this study, a suggested model for a smart parking system based on digital image processing was successfully tested and ran using many live video feeds from the produced prototype and a video spot of a parking lot from a drone. The technology accurately determines whether parking spaces are occupied by flashing a red outline if a car is within or occupying a parking place and then turning green when it is vacant. In addition, the suggested system allows the user to specify the borders of each parking place in relation to the camera's viewing angle by leveraging security video feeds from each parking lot, the system may be utilised to efficiently detect vacant parking spots dispersed over huge parking lots without incurring significant additional costs. The system was created using the Python and OpenCV platforms.

Future Scope

Safety precautions such as tracking the car number, facial recognition of the drivers, and automated charging can also be designed. We intend to expand the experiments to a real-time setting where consumers will be able to utilise the "Smart Parking" system on their portable devices. By using the Global Position Search (GPS) System to provide the path to the chosen parking area, the system may be improved even more. Future efforts to intelligently direct approaching automobiles to the nearest parking space and alert the driver if the parking lot is already occupied might be thought of as utilising machine learning algorithms and other computer vision techniques.

References

1. Y. Pankiv, N. Kunanets, O. Artemenko, N. Veretennikova and R. Nebesnyi, "Project of an Intelligent Recommender System for Parking Vehicles in Smart Cities," 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), 2021, pp. 419-422, doi: 10.1109/CSIT52700.2021.9648687.
2. Shaaban, K. and Tounsi, H., 2021. Parking detection method using quadtree decomposition analysis. Journal of Traffic and Transportation Engineering (English Edition).
3. D. K. Manase, Z. Zainuddin, S. Syarif and A. K. Jaya, "Car Detection in Roadside Parking for Smart Parking System Based on Image Processing," 2020 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2020, pp. 194-198, doi: 10.1109/ISITIA49792.2020.9163744.
4. S. Kazi, S. Khan, U. Ansari and D. Mane, "Smart Parking based System for smarter cities," 2018 International Conference on Smart City and Emerging Technology (ICSCET), 2018, pp. 1-5, doi: 10.1109/ICSCET.2018.8537281.
5. P. Melnyk, S. Djahel and F. Nait-Abdesselam, "Towards a Smart Parking Management System for Smart Cities," 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 542-546, doi: 10.1109/ISC246665.2019.9071740.
6. B. Kalaimathi, V. S. Charumathi, T. Aishwarya, M. A. Prasanna and S. Vijayakumar, "Raspberry PI Based Intelligent Car Parking System," 2021 Smart Technologies, Communication and Robotics (STCR), 2021, pp. 1-5, doi: 10.1109/STCR51658.2021.9588971.
7. M. Ramasamy, S. G. Solanki, E. Natarajan and T. M. Keat, "IoT Based Smart Parking System for Large Parking Lot," 2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation (ROMA), 2018, pp. 1-4, doi: 10.1109/ROMA46407.2018.8986731.
8. N. Sieck, C. Calpin and M. Almalag, "Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University," 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2020, pp. 1-6, doi: 10.1109/PerComWorkshops48775.2020.9156121.
9. I. M. Hakim, D. Christover and A. M. Jaya Marindra, "Implementation of an Image Processing based Smart Parking System using Haar-Cascade Method," 2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2019, pp. 222-227, doi: 10.1109/ISCAIE.2019.8743906.
10. M. Assim and A. Al-Omary, "A survey of IoT-based smart parking systems in smart cities," 3rd Smart Cities Symposium (SCS 2020), 2020, pp. 35-38, doi: 10.1049/icp.2021.0911.
11. ("Global Smart Parking Market Analysis Report 2022 Featuring Continental, Cubic, TKH Group, Cisco Systems, 3M, Deutsche Telekom, Vodafone, & Mindteck - ResearchAndMarkets.Com")
12. Koushika, B., PraveenKumar, G., Priyadharshini, S.T. and Balamurugan, A., 2021. An Efficient Car Parking Slot Identification System Using Image Processing. Annals of the Romanian Society for Cell Biology, pp.3400-3414.
13. Aashish Joshi , Arni Tharakaram Hariram , K M Vishall Somaiya , Mubashir Hussain, 2020, Smart Car Parking System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 09 (September 2020).
14. J. Ruili, W. Haocong, W. Han, E. O'Connell and S. McGrath, "Smart Parking System Using Image Processing and Artificial Intelligence," 2018 12th International Conference on Sensing Technology (ICST), 2018, pp. 232-235, doi: 10.1109/ICSensT.2018.8603590.
15. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.

16. Advani, V., 2020. 34 Open-Source Python Libraries You Should Know About [online]. GreatLearning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/open-source-python-libraries/ [Accessed 25 August 2022].
17. Anon, Introduction to NumPy [online]. Available at: https://www.w3schools.com/python/numpy/numpy_intro.asp [Accessed 26 August 2022].