A Project Work – Phase II on

Smart Parking Space Detection System

Submitted in partial fulfillment of the requirements for the award of the

Bachelor of Technology

in

Department of Computer Science and Engineering (Data Science)

by

Ajith Varma Jampana	**20241A6726**
Sreeshwan Jageer	**20241A6752**
Abhinav Sai Ratan Attemela	**20241A6701**

Under the Esteemed Guidance of

Dr. M. Kiran Kumar

Associate Professor

Department of Computer Science and Engineering (Data Science)

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Autonomous under JNTUH, Hyderabad)

Bachupally, Kukatpally, Hyderabad-500090

(2023-2024)

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Hyderabad-500090

CERTIFICATE

This is to certify that the major project (phase - II) entitled "**Smart Parking Space Detection System**" is submitted by **Ajith Varma Jampana (20241A6726), Sreeshwan Jageer (20241A6752) and Abhinav Sai Ratan Attemela (20241A6701)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science Engineering (Data Science) during Academic Year 2023-2024.

Internal Guide	**Head of the Department**
Dr. M. Kiran Kumar	**Dr. G. Karuna**
Associate Professor	**Professor**

External Examiner

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First we would like to express our deep gratitude towards our internal guide **Dr. M. Kiran Kumar,** Associate Professor**,** Department of Computer Science and Engineering (Data Science), for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. Sanjeeva Polepaka**, Associate Professor, Project Coordinator for timely suggestion, **Dr. G. Karuna,** Head of the Department**,** and to our principal **Dr. J. PRAVEEN,** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

Ajith Varma Jampana	**(20241A6726)**
Sreeshwan Jageer	**(20241A6752)**
Abhinav Sai Ratan Attemela	**(20241A6701)**

DECLARATION

We hereby declare that the major project (Phase – II) titled **"Smart Parking Space Detection System"** is the work done during the period from **12th January 2023 to 6th May 2024** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering (Data Science) from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad) . The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Ajith Varma Jampana	**(20241A6726)**
Sreeshwan Jageer	**(20241A6752)**
Abhinav Sai Ratan Attemela	**(20241A6701)**

ABSTRACT

The traditional process of finding parking spaces for vehicles can be time-consuming and hectic. It often leads to drivers circling around, wasting fuel, and contributing to traffic congestion. Therefore, there is a need for an automatic system that should address the challenges and optimizes the parking process for drivers. The automated system can be implemented using IOT. The proposed project (Parking Space Detection System) may revolutionize the way, parking spaces are located and utilized by leveraging the power of Opencv technology, which can assist the drivers to identify the parking space availability by providing real-time updates and navigation assistance. This system utilizes a combination of sensors, cameras, and AR-enabled mobile devices to create a seamless and intuitive parking experience. As a driver approaches a parking area, the system captures real-time data by bringing various pre-installed sensors in the parking area that includes occupancy sensors and cameras. The clever system then processes the collected data to locate and map open parking spots. Drivers may find the closest parking spot by using an Opencv application on their smartphone to overlay visual clues and directions onto their live video stream. Real-time information is provided by the Opencv interface, including parking space dimensions, the location of vacant spots, and navigation support.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

Internet of Things (IoT) allows data to be transferred over a network without the need for human contact. IoT enables users to move data into the cloud and makes affordable wireless technology available to them. IoT facilitates user transparency maintenance. The concept of the Internet of Things originated with the identification of devices to link. Computers connected to the internet can be used to monitor or control these devices. The terms "Internet" and "Things" are prominent in the acronym IoT. The Internet is a broad network that links computers and gadgets. Information can be sent, received, or even communicated with devices thanks to the internet. Traffic jams and air pollution are brought on by the parking issue. Today's population finds it difficult to find parking spaces in their daily lives. The latest survey predicts that by 2035, there will be a sharp rise in the number of vehicles—more than 1.6 billion. Every day, around a million barrels of oil are burned worldwide. As a result, the key to lowering the fuel waste stage is a smart parking system. The resolution to the issues that are brought up. Smart parking could be a way to reduce the amount of time and efficiency users spend looking for a spot, as well as the total cost of fuel used.

Modern technologies, such as image processing, computer vision, and sensor networks, are used by parking space detection systems to precisely identify and track the availability of parking spaces in real time. Authorities and drivers alike stand to gain from better overall urban mobility, less traffic congestion, and optimum parking utilization by utilizing these technologies. The way we approach urban mobility could be completely changed by the use of parking spot detection technologies. Drivers may make more educated judgments and spend less time looking for parking by having access to real-time information about available places, which also helps to improve traffic flow. Additionally, the incorporation of parking spot detection systems into smart city infrastructure promotes the growth of sustainable urban environments, reducing needless carbon emissions and raising inhabitants' and tourists' quality of life in general. Consequently, a smart parking system holds the key to reducing the fuel waste stage. The conclusion of the matters raised. The amount of time and effort drivers spend searching for a spot, as well as the overall cost of fuel consumed, may be decreased with the help of smart parking. . Drivers may make more educated judgments and spend less time looking for parking by having access to real-time information about available places, which also helps to improve traffic flow.

Integration of smart parking systems and augmented reality (AR) technology is a game-changer in the fields of urban mobility and parking management. How drivers navigate and interact with parking spots could be drastically altered by the use of augmented reality, which displays digital data onto the real environment.

Figure 1.1 General Parking System (Courtesy: Source [1])

Fig 1.1 Explains When augmented reality capabilities are combined with smart parking solutions, drivers can instantly acquire visual information about available parking spots, navigation routes, and relevant parking restrictions layered onto their perspective of the surrounding environment. In addition to making parking easier, this immersive and user-friendly method improves user experience overall, which boosts productivity, lessens traffic, and increases customer happiness. By combining AR technology with smart parking systems promoting data-driven decision-making and a more connected and sustainable urban ecology. AR-based smart parking systems make parking easier and less stressful for drivers by providing them with detailed, context-specific information. This improves environmental sustainability and urban mobility overall. However, the increase of more advanced technologies in the latter part of the 20th century laid the foundation for the evolution of smart parking systems as we know them today. An outline of significant turning points in the development of smart parking spot detection may be found here.

1.1 HISTORY

History of smart parking space detection can be traced back to the early 20th century, with the development of rudimentary parking meters and mechanical systems to regulate parking in urban areas. However, the increase of more advanced technologies in the latter part of the 20th century laid the foundation for the evolution of smart parking systems as we know them today. An outline of significant turning points in the development of smart parking spot detection may be found here.

Early Parking Meters, in the 1930s, the first parking meters were introduced in the United States, marking a significant shift in the management of parking spaces. These mechanical devices required drivers to pay for their parking time, providing a basic form of parking space regulation. The Black Maria, the first parking meter in history, was put in place in Oklahoma City, Oklahoma, in 1935. The initial purpose was to regulate parking and ensure turnover, thereby allowing more people to access businesses in the area. The early parking meters were mechanical devices operated by coins. They were set on a timer and would count down the allotted time for parking, typically an hour. Once the time elapsed, a metal "expired" flag would drop, indicating that the parking period was over. The standard fee was generally a few cents per hour.

Systems with sensors, Sensor-based parking systems, which use ultrasonic or magnetic sensors to determine if cars are in parking spaces, started to appear in the 1980s and 1990s. The foundation for more sophisticated sensor technologies found in contemporary smart parking systems was established by these early versions. Cities may lessen traffic congestion, increase motorist satisfaction, and optimize parking management tactics by utilizing the data gathered by these sensor-based systems. Drivers may make educated judgments about where to park by using digital signage and smartphone applications to receive real-time information about parking availability. These early iterations laid the groundwork for more advanced sensor technologies used in modern smart parking systems.

Integration of Computer Vision, with the advancement of computer vision technology in the 2000s, smart parking systems started incorporating sophisticated image processing techniques to identify and monitor parking space availability. This marked a significant shift towards more efficient and accurate detection methods. The integration of computer vision technology in parking systems has significantly transformed the way parking spaces are managed and utilized in urban environments. The artificial intelligence field of computer vision makes it possible for parking systems to evaluate and interpret visual data, resulting in more precise and effective detection of open parking spots. The field of smart parking has advanced in various ways as a result of this integration.

Integration of IoT and Wireless Communication, Smart parking systems have included Internet of Things (IoT) and wireless communication technology in the early 21st century. Through mobile applications and other digital platforms, customers can now get the most recent information about parking space availability thanks to this integration, which enabled for real-time data transmission. Parking space occupancy can be continuously monitored by wireless sensors and Internet of Things devices, which can then send the data to a central server. Drivers can swiftly and effectively locate available parking spaces by using this data, which is subsequently made available to them through digital signage and mobile applications. The addition of machine learning and data analytics algorithms has greatly enhanced the capabilities of smart parking space finding devices. By evaluating historical data and real-time patterns, these technologies can be used to maximize parking utilization, estimate the availability of parking spaces, and increase system efficiency levels. The system makes effective use of embedded internet and wireless devices, computers, sensors, object recognition, and image processing techniques.

In present scenario, the security gate is currently handled manually. By manually entering information on paper, all staff, students, and visitors' entries are maintained current. There are three authorized parking spots, P1, P2, and P3, which denote the number of spaces accessible for two- and four-wheelers, respectively. Each of the three spaces has a cement floor. P2 is employee unreserved parking, whereas P1 is divided into reserved and unreserved parking. These have marks on the edges of parking lots. Devices required to set up a smart parking system, such as cameras, sensors, and LEDs, can be mounted on the solar panel support structure. P3 is a region of clarity. There is not enough infrastructure to place devices like cameras and display panels. There is no bike parking signs at P1, P2, and P3. The idea for smart parking systems first surfaced as a way to address inefficiencies in parking in the late 20th century. As technology developed, systems that tracked spot occupancy with basic sensors were replaced by networked sensors, data analytics, and mobile applications that gave drivers access to real-time parking information. The advent of the internet and cellphones, which made it simple to get parking statistics and make reservations, accelerated this progress. In recent years, smart parking has combined with IoT, AI, and cloud computing to create incredibly complex and networked systems. At the moment, smart parking systems use a variety of technologies, including cameras, smart meters, and sensors, to maximize parking space utilization and improve urban mobility. Smart parking systems are positioned to have a significant impact on how transportation and urban development are developed in the future as cities deal with a growing parking problem.

1.2 CHALLENGES

Traditional parking systems face several significant challenges. Space utilization is a fundamental issue, as poorly designed parking lots or structures often lead to wasted space and inefficiency. This not only affects the overall capacity of the facility but also contributes to congestion and the inefficient use of valuable urban real estate. In high-demand areas, traditional parking facilities are prone to traffic congestion at entry and exit points, causing disruptions to the flow of surrounding traffic. The high cost of land acquisition in urban areas presents yet another challenge, making it difficult to provide affordable parking options for residents and visitors. Accessibility is an important issue because some conventional parking lots don't provide enough space for people with disabilities, which makes it difficult for them to park and use the neighboring services. Furthermore, depending too much on manual tasks for tasks like ticketing and payment processing might cause delays and annoyance for customers.

Smart parking systems come with their own set of challenges, Given the high upfront costs associated with the deployment of sensors, cameras, and software, many towns and enterprises find it difficult to afford the initial setup. The collection and archiving of data about vehicle movements and parking patterns raises concerns about data security and privacy. Software bugs or broken sensors are examples of technical issues that might cause the parking process to malfunction, which can annoy users and lower system reliability. Since integration with the current infrastructure occasionally necessitates the cooperation of multiple parties, including governmental bodies and technical businesses, it can be challenging. It might be difficult for users to adopt new technology since they may need to adjust their habits and behavior. The correct operation of smart parking systems depends on maintenance and upkeep, which can be expensive and need a specialized support infrastructure. Scalability is essential for these systems to be able to adjust to different parking lot kinds and increasing demand. Last but not least, depending on the region and jurisdiction, suppliers and operators of smart parking systems may find it difficult to handle the intricate web of rules, specifications, and legal requirements. New difficulties may arise when smart parking systems develop in the future.

How these systems will scale to support expanding metropolitan populations and rising automobile densities is one anticipated problem. As more cities adopt smart parking technologies, the demand for dependable and scalable solutions will increase, calling for ongoing innovation and infrastructural upgrades. Furthermore, the management and optimization of parking operations may become more challenging with the introduction of new technological advancements like artificial intelligence and machine learning. These technologies have the ability to boost productivity and make predictions, but they also require complex administration and maintenance to function optimally. Furthermore, as smart parking systems become more integrated with other aspects of urban infrastructure, such as traffic control and public transit networks, interoperability becomes increasingly vital. Optimizing the benefits of smart parking while reducing interference will need ongoing coordination and communication across multiple systems and stakeholders. Furthermore, as the automotive industry transitions to autonomous and electric vehicles, smart parking systems must adapt to their particular requirements. This means integrating and maneuvering autonomous vehicles in addition to building the infrastructure required for electric car charging stations. Furthermore, smart parking systems will encounter both opportunities and difficulties as urban mobility patterns continue to shift, including the emergence of shared mobility services and micro-mobility choices like e-scooters and bike-sharing programs. In order to provide equitable access to parking resources, future planning and implementation efforts for smart parking will need to include the diverse needs and preferences of various user groups. This will be a difficult but necessary task. To sum up, smart parking solutions may greatly increase urban mobility and reduce parking-related problems. However, in the years to come, they will also need to overcome persistent issues with scalability, technology integration, interoperability, and adaptation to changing urban mobility trends. It will need ongoing innovation, cooperation amongst stakeholders, and a progressive approach to urban design and transportation management to successfully handle these issues.

1.3 Working with OpenCV

An open-source library for image processing and computer vision applications is called OpenCV (Open Source Computer Vision Library). Since its initial development by Intel in 1999, a developer community has been actively maintaining and enhancing it. Because OpenCV is designed in C++ and has interfaces for Python, Java, MATLAB, and C++, it is widely usable in a variety of computer languages. The library offers a wide range of functions and algorithms for applications such as motion tracking, feature extraction, object detection and recognition, photo and video editing, and camera calibration, among others.Its broad toolkit makes it useful for many applications, including as robotics, augmented reality, medical imaging, driverless automobiles, and surveillance.The two primary benefits of OpenCV are its performance and flexibility. It is made to work well with a variety of platforms, such as desktop PCs, mobile devices, and embedded systems. Furthermore, programmers may quickly develop complex computer vision systems with OpenCV because of its easy interfaces with other frameworks and libraries, such as NumPy, TensorFlow, and PyTorch.

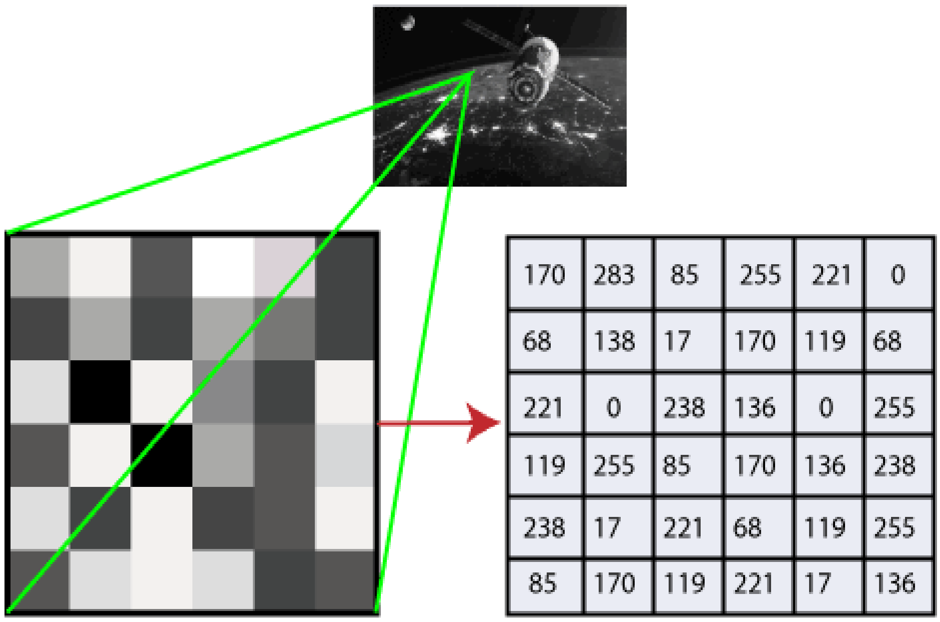

Figure 1.2 Basic OpenCV Working (Courtesy: Source [2])

Fig 1.2 Describes the model of Integrating OpenCV with parking space detection and vehicle detection involves leveraging the library's image processing and computer vision capabilities. Initially, preprocess live video feeds from cameras to enhance quality and reduce noise. Then, employ contour detection algorithms to identify parking spaces based on distinct boundaries or color variations. Analyze each space to determine occupancy status, possibly using background subtraction or pixel intensity analysis. Concurrently, extract moving objects from the video feed to detect vehicles, employing techniques like foreground extraction and object tracking. Optionally, integrate license plate recognition for further vehicle identification. Combine outputs to determine parking space availability, updating a database in real-time. Provide a user interface or API for users to access parking availability information and for system management. Optimize algorithms and parameters, testing in real-world conditions for accuracy and reliability. Ultimately, integrating OpenCV facilitates the creation of a robust smart parking system, efficiently managing parking space allocation and enhancing user experience.

By incorporating OpenCV into a smart parking system, other functionalities can be added in addition to the ability to recognize vehicles and parking spaces. For example, by using deep learning-based object detection models, the system can recognize cars with more accuracy, even in difficult situations like changing lighting or occlusions. Moreover, real-time analytics can be used to track how traffic moves around the parking lot, pinpointing regions of congestion and allocating parking spaces as efficiently as possible. Because of its flexibility, OpenCV may be integrated with other features, such anomaly detection to spot unregistered cars or questionable activity, or pedestrian detection to improve safety. Through constant improvement and extension of the system's functionalities using OpenCV, smart parking systems can adapt to the ever-changing demands of urban settings and offer effective. Distinct colored indications are typically used by automobile slot identification systems to show whether or not a car is parked in a particular slot. Here, red indicates that a car is parked in the slot, and green indicates that the space is either empty or that the car is not parked correctly. This visual cue expedites the parking process and reduces traffic by enabling drivers to discover available spaces quickly and simply. A car slot identification system uses a simple yet effective color scheme to immediately express the state of parking slots. The indicator light or display flashes green when a car takes up a slot, giving oncoming drivers looking for open spots a visual cue right away. On the other hand, the light turns green to indicate that a slot is empty or that there is a problem with a parked automobile when it is not correctly parked within it. By making it possible for drivers to immediately identify open spaces, this color-based approach reduces traffic and improves overall facility efficiency, which speeds up the parking process. Additionally, it encourages vehicles to reposition themselves in order to make better use of the parking area by drawing attention to inappropriate parking. Overall, the usage of both green and red signs improves user experience and streamlines traffic flow by providing drivers with a clear and simple guide for traversing parking lots.

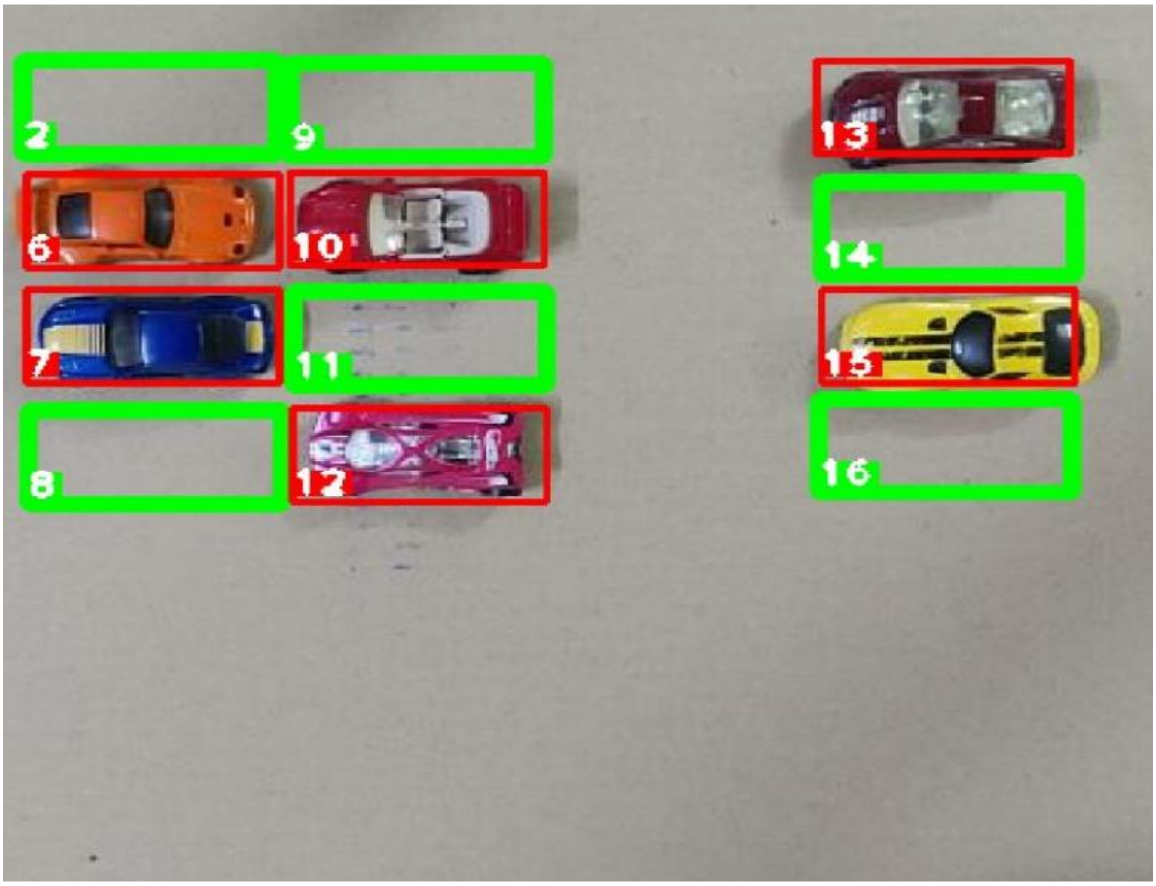

Figure 1.3 Slot Identification

Fig 1.3 Describes A parking slot identification system effectively communicates the status of parking slots by using a simple color scheme. Drivers looking for open parking may immediately see whether a car has taken a slot by looking at the red indicator. On the other hand, the indication changes to geen to indicate that a slot is empty or that there is an irregularity in parking when a car is parked there. By making it possible for cars to quickly find available spots, this color-coded system streamlines the parking procedure, lessening traffic and enhancing the general operation of parking facilities. Furthermore, it promotes healthy parking behavior by drawing attention to instances of inappropriate parking and incentivizing vehicles to reposition themselves in order to make better use of the parking space. In the end, the usage of both green and red indications together creates a more seamless and user-friendly navigation system for cars in parking lots, improving traffic flow and the overall parking experience. , it encourages vehicles to reposition themselves in order to make better use of the parking area by drawing attention to inappropriate parking. Overall, the usage of both green and red signs improves user experience and streamlines traffic flow by providing drivers with a clear and simple guide for traversing parking lots.

1.4 Objective of the Project

The aim of this project initiative will help drivers and parking facility managers by addressing a number of issues related to conventional parking systems. Reduce the overall costs of operating a parking facility by implementing the smart parking system that uses augmented reality and sensors and radars in the internet of things (IOT). Can employ sensors to determine when a spot is empty and release it for usage by other cars, saving labor-intensive tasks. Additionally, make the parking system more effective so that cars can find the closest spot that is open while also saving time and gasoline.

1.5 Methodology

To achieve Smart Parking Space Detection system is used in conjunction with image processing, computer vision, and artificial intelligence techniques. It entails gathering real-time parking data and creating an easy-to-use open-source video application that superimposes it on users' displays. The system guarantees constant updates and user engagement while offering real-time navigation support and parking space availability. It seeks integration with current infrastructure and smart city initiatives, encourages user feedback and data sharing to enable continuous improvement, and offers a comprehensive solution to mitigate urban parking issues and create a more effective and user-friendly urban transportation experience. An augmented reality interface is created, providing users with a customizable interface and 3D cues for easy navigation. . By making it possible for cars to quickly find available spots, this color-coded system streamlines the parking procedure, lessening traffic and enhancing the general operation of parking facilities. While customers may easily book and pay for spots. Facility improvement can be aided by data analytics, which also offer comprehensive user support and training. The parking facility is where the system is installed, and regular maintenance makes sure it runs well. In conclusion, this approach aims to transform parking by offering a quick, user-friendly, and secure parking

1.6 Architecture Diagram

An ecosystem with multiple facets makes up the architecture of an OpenCV smart parking space detecting system. Parking lots with sensors installed are the first step, as they continuously gather data on available parking spaces in real time. User devices like smartphones and AR glasses are part of the AR system, which processes and integrates this data. The Opencv system uses object identification and computer vision to build spatial maps of the building and precisely overlay real-time parking spot availability data, which includes navigational aids and 3D hints. Customization options and accessibility features enhance user experience, while obstacle detection technology ensures safety. Reservations are seamlessly integrated, and data analytics provide insights for facility optimization. Regular software updates maintain the system's relevance, and thorough testing, user training, and ongoing maintenance complete the architectural framework. This ecosystem revolutionizes parking by offering a user-friendly, efficient, and data-driven parking experience, benefitting both drivers and facility operators.

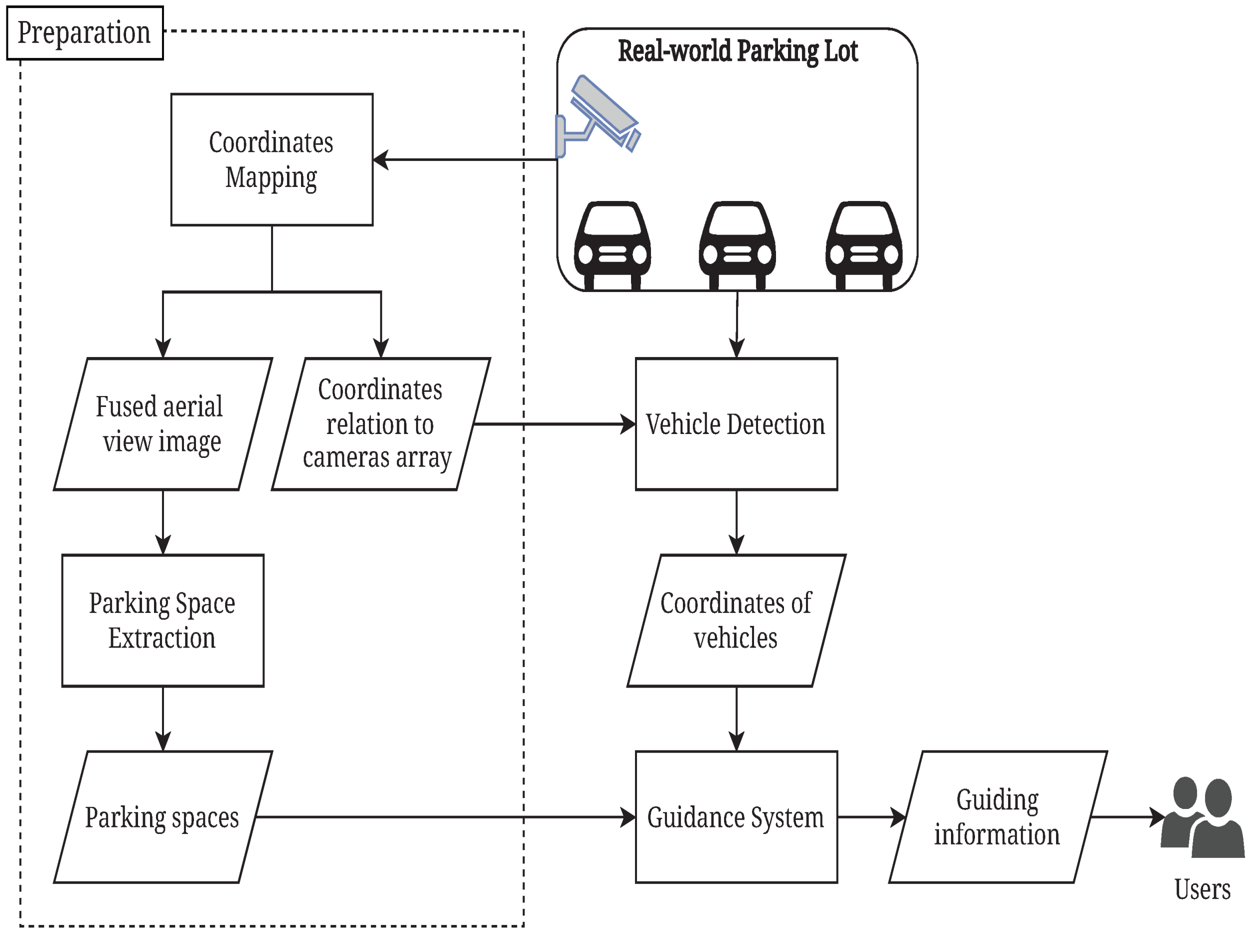

Figure 1.3 Architecture Diagram

1.7 Organization of the Report

This report consists of the overview of what all topics discussed in this entire report in a brief and concise manner with the sequence of topics presented. The architecture is described in Fig 1.3

Chapter 1: Introduction

This chapter initiates with an exposition on the project's contextual background, progressing to a delineation of the defined objectives and the methodological framework employed to fulfill these objectives. An architectural diagram is presented, serving as a visual representation of the project's intricacies.

Chapter 2: Literature Survey

This chapter functions as an exhaustive exploration of existing research within the project's domain. It systematically reviews and analyzes methodologies from 15 research papers, elucidating the results attained through their respective approaches. Additionally, a comprehensive evaluation of each paper's merits and drawbacks is provided, contributing to a nuanced comprehension of the current landscape.

Chapter 3: Proposed Method

The core of the document, this chapter commences with a succinct articulation of the problem statement, expounding on the project's objectives. The architecture diagram is delineated, providing a thorough overview of the structure and interconnections between various modules. A modules connectivity diagram further enhances clarity. Software and hardware requisites are meticulously detailed, ensuring a precise understanding of the project's technological infrastructure. The chapter then transitions to an exhaustive description of individual modules, encompassing functional and non-functional requirements. To facilitate understanding, visual aids such as class, sequence, activity, and use case diagrams are incorporated.

Chapter 4: Results and Discussion

This chapter delves into the practical facets of the project, commencing with a comprehensive elucidation of the dataset employed. Experimental results are presented, accompanied by supportive plots illustrating outcomes effectively. The advantages of the proposed work are discussed, offering insights into the positive aspects of the project. The chapter subsequently explores the significance of these results, establishing the relevance and impact of the project within its domain.

Chapter 5: Conclusion and Future Enhancements

Summarizing the project's trajectory, this chapter revisits initial objectives, the approach taken, the project's importance, and the results obtained. It offers a holistic view of the project's accomplishments and their implications. Furthermore, the chapter provides a glimpse into potential future enhancements, outlining the trajectory for future development and improvements.

Chapter 6: Appendices

The final chapter provides supplementary resources to augment the reader's understanding. Code samples from the project are included in this section, serving as a practical reference for those interested in the technical intricacies of the work. This appendix functions as a valuable supplement to the main body of the document, offering hands-on insights into the project's implementation.

CHAPTER – 2

LITERATURE SURVEY

2.1 Existing Approaches

Gaurav Satyanath and Jajati Keshari Sahoob (2020) [3] proposed for dehazing networks to maximize the performance of the system on both hazy and non-hazy conditions. The suggested technique can be implemented into current smart parking systems, which use a small number of cameras to keep an eye on hundreds of parking spots. The author used the task-driven, real-world test set of the RESIDE-beta dataset to create a custom foggy parking system dataset in order to validate this methodology. Using CNRPark-EXT and fuzzy parking system datasets, the suggested method is evaluated against the most advanced parking space detectors currently available. The proposed approach has a significant improvement in accuracy on the foggy parking system dataset, according to experimental results.

Rifath Mahmud and Saifuddin Saif (2021) [4] the goal of this study is to compare and thoroughly examine earlier studies on the detection of vacant parking spaces from various angles. Methods from earlier studies are described in detail, along with their benefits and drawbacks. The dataset, accuracy, processing time, and other performance metrics are compared with the experimental findings. Six generalized stages were used to compare the frameworks of previous investigations. This paper also addresses the challenges associated with vision-based unoccupied parking space recognition, which will be useful for future research and present an interesting problem for scientists to solve.

Purva Hattale and Vikrant Jangam suggested (2019) [5] explains that as the number of vehicles grows in the current day, so does the difficulty of parking. The author of this research suggests parking-space occupancy detection. Display of available parking spots, parking data, Wireless connectivity, readily accessible parts, the parking lot's live video feed from the camera will be received by the system. Every time a car pulls into or out of the parking lot, pictures are taken. The system is also mobile phone (browser) compatible.

This Authors Suggested Ruidi yang and xiangwu (2021) [6] the study applies vehicle detection in parking lots and parking spaces, utilizing the YOLO v3 network. In order to enable deep networks to extract more fine-grained features, this article, which is based on YOLO v3, adds a residual structure to extract deep car parking space characteristics and uses four distinct scale feature maps for object detection. The findings of the experiment demonstrate that this strategy can lower the rate of missed detections while increasing the accuracy of car and parking spot detection.

Authors ElakyaR and Juhi Seth (2020) [7] explains a clever and effective method for automating the parking system's management, which uses internet of things technology to assign a parking place efficiently. The Internet of Things (IoT) offers wireless connectivity to the system in this case, allowing the user to monitor parking spot availability. As the number of automobiles in metropolitan areas increases, road congestion is becoming a serious issue. The user receives a notification with the parking information. As a result, the user's waiting time while looking for a parking spot is reduced. Car theft is being prevented with the use of RFID technology.

Xumin Huang, Peichun Li (2021) [8] proposed Smart Parking system consists of an IoT module is deployed on-site as part of the proposed smart parking system, and it is utilized to track and indicate the availability of each individual parking space. The Long Short-Term Memory (LSTM) model is utilized by the Parked Vehicle Assisted Edge Computing (PVEC) system to estimate parking spaces. Using LSTM and federated learning to encourage PLO cooperation in parking space estimation.

Daniel and Miguel (2021) The suggested methodology uses the COCO model for object detection and the YOLO-v5 [9] for vehicle detection. The dataset for training and validation is called PKLOT. The obtained accuracy is 98.9%. The YOLO v3 network is the foundation of this article, which is applied to parking spots and vehicle detection in parking lots. This work, which is based on YOLO v3, employs four distinct scale feature maps for object recognition and adds a residual structure to extract deep car parking space features.

Authors Lou, Qiang (2020) [10] This research proposes an IoT-driven vehicle detection method that blends the heterogeneous data feature of the UWB channel with that of the magnetic signal, suggesting the methodology increase the performance of the WVDs. The channel impulse response (CIR) signature and propagation path length—both of which can be obtained from UWB modules—are presented in the suggested method in order to achieve vehicle detection. The suggested approach was tested in several settings in a business parking lot. The tests demonstrate that at a sampling rate of 1 Hz for the magnetic sensor, it has a detection accuracy of 98.81%.

The Study conducted by Varghese and G. Sreelekha (2019) [11] suggests a machine learning method for automatically identifying empty lots in parking spaces that are delimited, meaning that the limits of each individual lot are established. The method is then extended to encompass non-delimited parking places in a wider range of situations. Because it can train the classifier with fewer images than state-of-the-art approaches, handle non-rectangular, variable-sized photographs as input, and be used to non-delimited parking spaces, the suggested method is appropriate for real-world deployment.

The Study conducted by Zhanghua and Yantao (2021) [12] suggests creating a high-gain dielectric lens using 3D printing A smart parking radar system that uses a magnetic sensor for dual-mode detection to increase parking spot detection accuracy is developed to use a K-band antenna. The main feed antenna for the suggested antenna is a wideband high-efficiency electromagnetic structure (WHEMS). A conformal flat semi closed perforation dielectric lens is made to attain a high gain while avoiding the measurement inaccuracy brought on by excessive penetration losses in the K-band. This lens takes advantage of low-cost, quick prototyping 3-D printing technology. Operating at 24-24.25 GHz, the K-band radar has a measured gain of 18.4 dBi.

The research by Yuanfeng and Boya (2013) [13It is said that in order to improve vehicle detection accuracy in a variety of illumination conditions, they created an adaptive multimodal feature fusion and cross-modal vehicle index (AFFCM) model. AFFCM is based on the single-stage object detection paradigm and uses RGB and TIR images. The proposed method uses a multiscale feature pyramid to achieve adaptive fusion. A cross-modal vehicle index is developed in order to minimize false alarms in vehicle detection, extract the target region, and suppress complicated background information. On the Drone Vehicle dataset, the mean average precision (mAP) is 14.44% and 5.02% higher than the values obtained with just RGB or TIR images. The mAP is 2.63% higher than state-of-the-art methods that employ RGB and TIR images.

This work Proposed by Nikhil and vineetha (2020) [14] intends to carry out an extensive analysis of the parking infrastructures that are currently in place and suggests clever parking solutions that make use of cutting-edge Big Data Analytics and Deep Learning methods. We anticipate that adding spatial-temporal information will improve CNNs' classification capabilities. The author plans to develop a parking-assistance mobile application for iOS and Android that displays a real-time grid layout of available and occupied parking spots using the user's GPS data. The parking information from the cloud-classifier will be used to update this app on a regular basis. The study's findings ought to facilitate customers in finding the nearest parking space and improving their time management. Better parking management services and more efficient traffic flow are also expected.

In this paper the authors Chin, Nyean (2017) [15] recommended an outside parking spot emptiness detection machine that uses cellular gadgets to provide drivers of cars greater statistics approximately parking areas, such as their place and occupancy. Using a Raspberry Pi and a cutting-edge picture popularity algorithm, Convolutional Neural Network, the machine acknowledges open parking areas from a actual-time picture of the automobile parking space captured via way of means of an IP camera. To take a look at the recommended technique for actual-time parking spot emptiness detection, a college automobile parking space has been selected. The Driver App is an Android phone software designed to offer drivers with ubiquitous show of occupancy statistics for out of doors parking spots in actual time. Based on the responses to the System Usability Scale (SUS) questionnaire, the evaluation results showed that the Driver App is a highly usable application that offers smart parking services to help drivers find available parking spaces.

Unoccupied parking space detection systems are becoming increasingly important because one of the biggest problems for drivers in urban areas is avoiding traffic congestion and the tedious chore of searching for an empty parking space. In order to get over these challenges, the scientists suggested a method based on a Dilated Convolutional Neural Network that was created especially for identifying.parking space occupancy in a parking lot, given only an image of a single parking spot as input. The proposed method Shezod and woong (2019) [16] demonstrates greater reliability than earlier research, particularly when tested with a whole different set of photos. We also evaluated our model in comparison with AlexNet, as in earlier research, the approaches' performance was evaluated against the well-known architecture, AlexNet, which exhibits a very promising accomplishment.

In metropolitan locations of both wealthy and developing nations, parking is a big issue. The proposed solution Vladimir and Josef (2020) [17] makes use of contemporary IoT techniques and technologies like IQRF, sensors, and tiny PC platforms. It's an affordable alternative to a specialized, high-priced system that can be expanded and integrated with other Internet of Things services.

Poor parking infrastructure leads to a number of problems, including increased automobile emissions, traffic congestion, and substantial driving time wasting. This issue may be resolved by using smart technology to make this process easier. This will increase operational effectiveness, streamline urban traffic flow, and provide drivers with a more convenient and pleasurable driving experience. The proposed method by Khanna and Anand (2016) [18] IoT-based smart parking system allows real-time data about parking availability to be obtained via a mobile application. There is an Internet of Things device with sensors and microcontrollers in every parking spot. The device will send its data to a cloud-based system, which will use it to analyze and determine how many parking spaces are available with open slots. Users are provided with up-to-date information regarding the availability of every parking space, enabling them to select their preferred option. The outcomes demonstrate that, on our dataset, our method can increase generic object detection accuracy by 11%.

Mohamad and bilal (2020) [19] proposed a system that creates an augmented reality learning environment while accounting for several instructional methodologies, such as collaborative learning. Students can engage with the course material and other students concurrently through collaborative learning, which promotes deeper understanding and more motivation. Several educational approaches, such as collaborative learning, were taken into consideration when developing an AR learning environment. Students can engage with the course material and other students concurrently through collaborative learning, which promotes deeper understanding and more motivation.

There is a lot of research being done on augmented reality. It is one of the few concepts that, although being seen to be unrealistic and unachievable in the past, can be used somewhat successfully today. Many universities and high-tech businesses are still in the early phases of their AR research and development. Numerous experts anticipate that it will rank among the most widely used technology in the future, much like our computers and smartphones. This paper by Urvashi (2022) [20] provides a comprehensive study of AR including its working, applications, current challenges and future trends.

Xiang and yixu (2020) [21] suggested a unique method to enhance object detection on mobile devices by fusing semantic data from object detectors with geometric information from VIO. Our system consists of three parts: an online semantic map, a scale-based filtering strategy, and an image orientation correction method. Every element capitalizes on the many features of the VIO-based AR framework. We manually classified items in image sequences from 12 room-scale AR sessions in order to validate our methodology. The outcomes demonstrate that, on our dataset, our method can achieve a 12% improvement in accuracy over generic object detectors.

The author Yaniel gu (2021) [22] exhibited a method for indoor navigation based on AR. The suggested system creates a point cloud map using Simultaneous Localization and Mapping (SLAM) and carries out navigation and placement in a unique hybrid map that combines the indoor environment's floor plan and 3D point cloud map. According to the experimental findings, the suggested AR-based navigation system can function satisfactorily when it comes to first-person view navigation. The outcomes demonstrate that, on our dataset, our method can increase generic object detection accuracy by 17%.

In this paper the authors Chandra and michal (2018) [23] Presented The integration of IEEE 802.15.4 Wireless Sensor Network (WSN), Automatic License Plate Recognition (ALPR), and Radio Frequency Identification (RFID) technologies is the basis for the proposed Smart Parking Management System (SPMS). The system is able to gather data on parking spot occupancy status and send it to a database server via ALPR and WSN. Users can obtain real-time updates by using a mobile application to access this data. Additionally, the program supports online payment processing and parking spot reservations, as well as NFC-based user identification. A deployed, tested, and validated proof-of-concept has been made. The validation shows the suggested system's ability to fulfill the actual SPMS requirements.

The research by Madhuri and Robin (2020) [24] tackles the issues at hand by combining computer vision, IoT, and mobile applications to solve them. If put into practice, this approach will undoubtedly save a significant amount of time. Authors may also use automatic security bollards to ensure the safety of the parked car. Ultrasonic sensors will be used as proximity sensors, while Node MCU will serve as the microcontroller. Additionally, in order to remove all false positives, the author will validate readings from IoT devices using live CCTV video. All users of the mobile application will always see the real-time status of the parking spots in the parking lots thanks to the system.

The proposed scheme by yang and yanto (2012) [25] comprise mobile phone applications, embedded web servers, central web servers, and wireless sensor networks. Every parking space in the system has a low-cost wireless sensor network module installed, and each module has one sensor node. Sensor nodes monitor parking space conditions, and they provide periodic reports to the embedded web server via installed wireless sensor networks. Real-time data is transmitted to a central web server via Wi-Fi networks, and regular mobile devices are also used by the driver of the car to locate empty parking spaces.

Convolutional Bidirectional LSTM models serve as the foundation for the suggested approach by Poala and Stefano (2023) [26]. The outcomes have been compared using precise data based on both occupancy and free slots. The paper also provides a framework for converting models that rely on open slots to occupancy-based evaluation models and vice versa. The obtained results have improved upon those already published in the literature. The Snap4city platform and infrastructure were used to gather data from garages in the Florence, Tuscany, Italy area for the comparison.

The research by Kamal and Stephen (2018) [27] proposes a smart car parking system that will help users identify parking spots and cut down on the amount of time they spend looking for the closest parking lot. In order to minimize the amount of data transferred over the network, the suggested system gathers the raw data locally and applies data filtering and fusion algorithms to extract features. Subsequently, machine learning algorithms are used to process and evaluate the converted data on the cloud. Additionally, in order to remove all false positives, the author will validate readings from IoT devices using live CCTV footage. All users of the mobile application will always see the real-time status of the parking spaces in the parking lots thanks to the system.

2.2 SUMMARY OF EXISTING APPROACHES

To define the problem statement for the project, various literature reviews and existing approaches have been revised. The summary of the existing approaches are shown in below Table 1.

Table 2.1. Summary of Existing Approaches

Reference Number	Paper Name	Author & Year	Methodology Adapted	Remarks/Results
[3]	Smart Parking Space Detection under Hazy conditions using Convolutional Neural Networks	Gaurav Satyanath and Jajati Keshari Sahoob /[2020]	Approach using convolution neural network Methodologies used in this paper were mAlexnet (a cnn	The objective was to provide parking space in hazy conditions Accuracy is 80.37% on hazy parking

			model)	system
[4]	A Comprehensive Study of Real-Time Vacant Parking Space Detection Towards the need of a Robust Model	Rifath Mahmud and Saifuddin Saif /[2021]	Nueral network and vector machines (svm) are used to develop this model.	Dataset used is PKLot Dataset PKLot consists of 695,900 images Accuracy is 86.7%.
[5]	Parking Space Detection Using Image Processing	Purva Hattale and Vikrant Jangam s /[2019]	The cv2 module is used for image capturing and dnn algorithm for image preprocessing	Dataset used is PKLot Dataset consists of 695,900 images
[6]	Vehicle and Parking Space Detection Based on Improved YOLO Network Model	Ruidi Yang and Xiangwu Ding /[2021]	Approach using Resnet neural network Methodologies used in this paper were Yolo v3 network and Darknet 53	The PKLot, COCO, and PASCAL VOC datasets were used, and the accuracy rate was 87.2%.
[7]	Smart Parking System using IOT	ElakyaR and Juhi Seth /[2020]	The hardware components are RFID Sensor,GSM module and IR sensors	The cloud database is used .Accuracy is 91.7%.
[8]	A Federated Learning based Parking Space Estimation with Parked Vehicle assisted Edge Computing	Xumin Huang, Peichun Li / [2021]	For estimating parking spaces, the Long Short-Term Memory (LSTM) model is used by the Parked Vehicle Assisted Edge Computing (PVEC) system.	Federated learning and LSTM adoption to encourage PLO cooperation in parking space estimate.
[9]	T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural	Daniel, Miguel /[2021]	Vehicle detection is done with YOLO-v5. For object detection, the	The dataset utilized for both validation and training is PKLot. and 98.9%

	Networks		COCO model is employed.	accuracy was attained.
[10]	An IoT-Driven Vehicle Detection Method Based on Multi-source Data Fusion Technology for Smart Parking Management System	Lou, Qiang /[2020]	The algorithm for vehicle detection based on UWB is employed. It is suggested to combine UWB with magnetic technology in order to increase WVD (wireless vehicle detection) accuracy.	Accuracy is 98.81%
[11]	An Efficient Algorithm for Detection of Vacant Spaces in Delimited and Non-Delimited Parking Lots	Varghese and G. Sreelekha /[2019]	The PKLot dataset is introduced to evaluate two textural descriptors, Local Binary Patterns	A single svm classifier to discriminate between occupied and vacant lots
[12]	3D Printing Conformal K-band Lens Antenna for a Smart Parking Space Detection System	Zhanghua,Yantao /[2020]	The primary feed antenna for the antenna is a wide-band high-efficiency electromagnetic structure (WHEMS).	Complex microwave components, including as measurement systems, reflection arrays, absorbers, and antennas, have been employed for vehicle identification based on 3D printing technology.

[13]	Vehicle Detection Based on Adaptive Multimodal Feature Fusion and Cross-Modal Vehicle Index Using RGB-T Images	Yuanfeng,Boya /[2023]	It is suggested to use a multimodal adaptive feature fusion (MAFF) module. It is suggested to use the soft-pooling channel attention (SCA) approach to extract notable features from RGB-T images in various lighting scenarios.	On the Drone Vehicle dataset, mean average precision (mAP) is 14.44% and 5.02%. And 76.5% accuracy was attained. The dataset consists of 12964 photos taken from three distinct angles and 15475 image pairs that were acquired vertically.
[14]	A Smart Eco-System for Parking Detection Using Deep Learning and Big Data Analytics	Nikhil, Vineetha /[2020]	A wide range of knowledge discovery is offered by the proposed CNN framework FMR-CNN, which makes use of the Mask-RCNN version.Proposed CNN framework FMR-CNN that uses the Mask-RCNN version provides a broad spectrum of knowledge discovery	Dataset-PKLot from Brazil's Universidad Federal do Parana A parking assistance smartphone application for Android and iOS that uses the user's GPS data to show the grid structure of available and occupied parking spaces in real time
[15]	Mobile Outdoor Parking Space Detection Application	Chin, Nyean /[2017]	Utilizing a Raspberry Pi and a state-of-the-art image recognition algorithm,	The Driver App is an Android smartphone application

			Convolutional Neural Network, the system is able to detect open parking spots from a real-time image of the parking lot obtained by an IP camera.	designed to allow drivers of vehicles to view real-time occupancy statistics for outdoor parking spaces everywhere they go.
[16]	Generalized Parking Occupancy Analysis Based on Dilated Convolutional Neural Network	Sherzod, Woong /[2019]	For image processing and detection, dilated convolution layer and alexnet are employed. CarNet design is based on dilated convolutional layers and is motivated by our objective, which is to detect automobile occupancy in a parking lot.	The 12,417 photos of parking lots that make up the PKLot dataset are used. The dataset is divided into three subsets: UFPR04, UFPR05, and PUCPR. These subsets were manually classified into subfolders labeled as sunny, overcast, or rainy based on the observed weather.
[17]	A Smart Parking System Based on Mini PC Platform and Mobile Application for Parking Space Detection	Vladimir, Josef/[2020]	The suggested parking system, which is built on the small PC platform, is an intricate and cost-effective method for locating open spots in an outdoor parking lot.	Identification of Vehicles. The MPU-9250 magneto metric sensor, which is coupled to an Arduino Mini microprocessor unit via an I2C

				interface, is used to identify vehicles.
[18]	IoT based smart parking system	A. Khanna, R. Anand /[2016]	The Mask-RCNN variant of the proposed CNN architecture, FMR-CNN, offers a wide range of knowledge discovery.	Brazilian university Federal do Parana's dataset, PKLot.
[19]	Collaborative Augmented Reality in Education: A Review	Mohamad bilal /[2020]	This essay examines the uses of augmented reality in the classroom and talks about how it affects learning objectives and student participation.	Increased learning outcomes and increased student engagement .
[20]	Augmented Reality Technology: Current Applications, Challenges and its Future	Urvashi, /[2022]	This paper explores techniques to improve user interaction in augmented reality gaming, focusing on gesture recognition and feedback mechanisms.	User studies, gesture recognition algorithms, and usability testing.
[21]	Object Detection in the Context of Mobile Augmented Reality	xiang, yixu /[2020]	The paper presents a real-time object recognition system for augmented reality applications, highlighting its accuracy and efficiency.	Computer vision techniques, machine learning algorithms, and benchmarking against existing methods. And 95% accuracy in object recognition, low latency.

[22]	AR-based Navigation Using Hybrid Map	Yaniel Gu /[2021]	An AR-based indoor navigation system is presented in this paper. The suggested system creates a point cloud map using Simultaneous Localization and Mapping (SLAM) and carries out navigation and placement in a unique hybrid map that combines the indoor environment's floor plan and 3D point cloud map.	The testing findings showed that first-person view navigation can be satisfactorily provided by the suggested AR-based navigation system.
[23]	Smart Parking Management System: An integration of RFID, ALPR, and WSN	Chandra, Michal/[2018]	The system may gather data regarding parking space occupancy status using ALPR and WSN, then send it to a database server. Users can obtain real-time updates on this information by using a mobile application.	Additionally, the program promotes NFC-based user authentication and makes online payment and parking space reservations easier for users. There is a deployed, tested, and verified proof-of-concept.
[24]	Smart Parking with Computer Vision and IoT Technology	Madhuri, Robin /[2020]	To address these issues, the author suggested combining computer vision, IoT, and mobile applications. If this is done, it will undoubtedly save a lot	All users of the mobile application will always be able to see the real-time status of the parking spaces in the

			of time. Additionally, we will be able to use automatic security bollards to ensure the safety of the parked car.	parking lots thanks to the system. .
[25]	Smart parking service based on Wireless Sensor Networks	Yang, Yanto /[2012]	The proposed system consists of wireless sensor networks, a mobile phone application, an embedded web server, and a central web server. A cheap wireless sensor network module with one sensor node is deployed in each parking area in the system.	Wi-Fi networks are utilized to send real-time data to a central web server, and drivers can also use standard mobile devices to find vacant parking lots.
[26]	Predicting Free Parking Slots via Deep Learning in Short-Mid Terms Explaining Temporal Impact of Features	poala,Stefano /[2023]	The Convolutional Bidirectional LSTM models provide the foundation of the suggested approach. Every 15 minutes, this study addresses deep learning model solutions based on a 24-hour mid-term prediction.	Precision measurements based on both occupancy and free slots have been used to compare the results. Additionally, the study offers a framework for switching from occupancy-based assessment models to models that rely

				on open slots, and vice versa.
[27]	Smart Car Parking System Solution for the Internet of Things in Smart Cities	Kamal, Stephen /[2018]	In order to help users identify parking spaces and reduce the amount of time they spend looking for the closest parking lot, the study suggests a smart auto parking system.	In order to minimize the amount of data transferred over the network, the suggested system gathers the raw data locally and applies data filtering and fusion algorithms to extract features. Following that, machine learning techniques are used to process the converted data on the cloud.

CHAPTER – 3

PROPOSED METHOD

3.1 PROBLEM STATEMENT

The traditional process of finding parking spaces can be time-consuming, frustrating, and inefficient. It often leads to drivers circling around, wasting fuel, and contributing to traffic congestion. Therefore, there is a need for an IoT solution that should address the challenges and optimizes the parking process for drivers. Parking Space Detection using IoT System and Augmented Reality may revolutionize the way, parking spaces are located and utilized by leveraging the power of AR technology. Which can assist the drivers for identifying the parking space availability by providing real-time updates, navigation assistance.

3.2 OBJECTIVES

The proposed work has five objectives and are listed below:

- To implement the smart parking system using Camera and radors in internet of things (IOT)
- To reduce the overall costs associated with running a parking facility. For example, you can use sensors to detect when a space is vacated and open up that space to other vehicles, reducing the need for manual labor
- To increase efficiency in parking system for the drivers for directing to the nearest available parking spot and helping to save fuel and time
- To embed the augmented reality guidance system using sensors and on-board screen data.
- To detect the nearest parking slot for increasing efficiency which saves fuel and time.

3.3 PROPOSED METHODOLOGY

This section includes the implementation of proposed work and its modules.

1. Implementation of proposed work
2. Modules and its Description

3.3.1 Implementation of Model

The below diagram shows the implementation of the model.

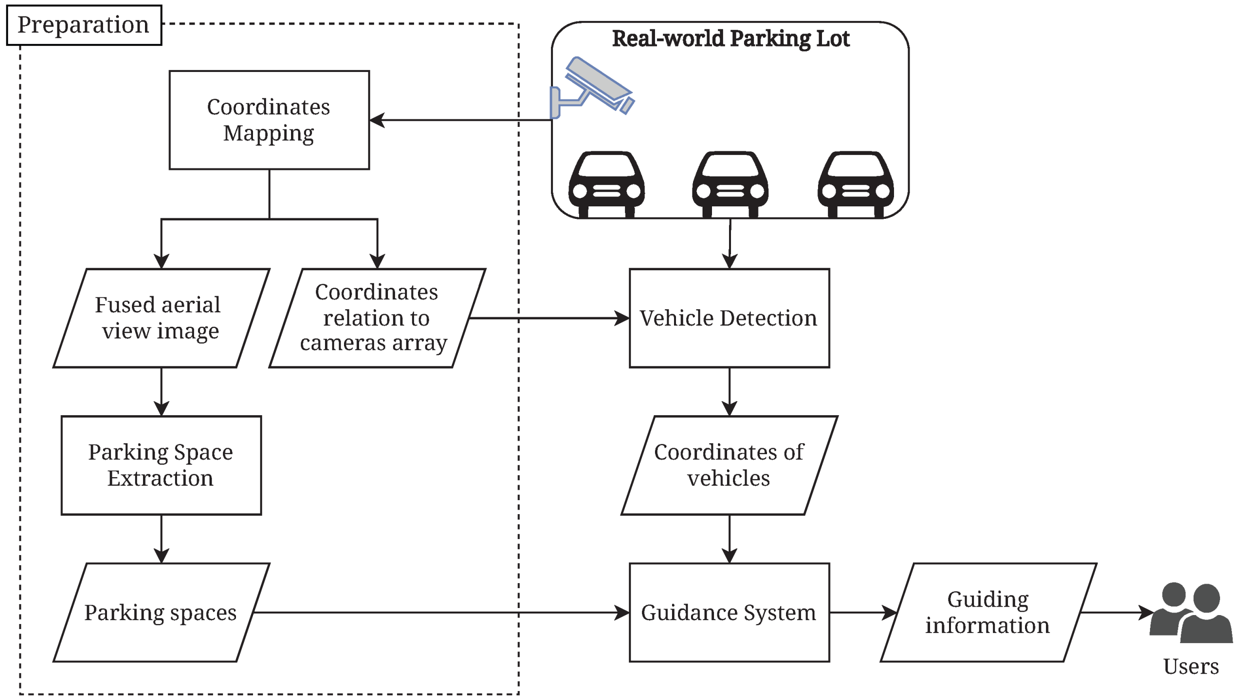

Figure. 3.1 Smart Parking System Architecture Diagram

The Figure 3.1 represents the Architecture of proposed Smart Parking System. Here is a general overview of how the application works:

Smart Parking

One of the numerous technologies included in a smart city concept is smart parking. Every gadget in the parking lot is connected to an electrical device. An Internet of Things is created by connecting this integrated gadget to the internet.

Location-Based Service (LBS)

LBS is a software-oriented information service that mobile devices offer to carry out control activities based on location information. Through the internet network, a mobile device can access this information service, which can show the device's location. The LBS service used in this study provides data on the path followed by those using two wheels to get to the parking lot.

Internet of Things (IoT)

An intelligent and interoperable node connected to a dynamic worldwide network architecture is what is referred to as the Internet of Things. IoT utilizes any notion of connectivity at any time and from any location. IoT, to put it simply, is the capacity to intelligently connect everything in our environment to the internet while accounting for the existence of some degree of privacy and autonomy. An item or thing mentioned in this conversation

Hardware Design

For the device to send data to Firebase, an internet connection is also necessary. The Raspberry pie will be the hardware component that we use. The microcontroller receives the data that was read from the Raspberry Pie. The microcontroller has been programmed to enable user entry into the parking area.

Location-Based Service (LBS) Design

In order to install Location-Based Services (LBS) on a smartphone, the user must be able to access GPS services and have an internet connection. Users can obtain trip route information to the chosen parking place upon the completion of these two requirements.

Application Design

The Parking Points application is made with multiple features that users can utilize to find the location of the assigned parking place. The program is integrated with multiple other support systems, including the Google Maps application, the Firebase real-time database, and the Google Maps API.

Camera Integration and preprocessing

Several critical measures are done in OpenCV's preprocessing stage to improve detection accuracy when detecting possible car zones. To speed up processing and decrease computing complexity, the input photographs are first transformed to grayscale. Subsequently, a blur filter, such as Gaussian blur, is used to sharpen edges and reduce noise, allowing for better distinction of vehicle components. Following that, thresholding algorithms divide the picture into binary zones based on pixel intensity levels, separating objects from the background. Contour identification algorithms are then used to the binary picture to find prospective automotive areas, with size, aspect ratio, and other features serving as criteria to select contours that resemble probable car formations. Concurrently, during camera sensor integration, the camera's clocks are synced to catch and hold charge, with the readout electronics prescribing the integration's limits independent of shutter exposure. This step optimizes sensor performance by synchronizing clock settings for effective charge trapping. Furthermore, size, aspect ratio, and other characteristics are used to distinguish forms matching possible automotive designs, assisting in accurate detection. This integration period is critical for increasing sensor sensitivity and reducing noise, hence improving the overall quality of collected photos.

3.3.2 Software and Hardware Requirements

To improve the performance of the project, following software and hardware requirements are needed:

Software Requirements:

Table 3.1. Software Requirements

Software Product	Specification	Purpose
OpenCv	4.9.0	For vehicle identification
Arduino IDE	5.3	For execution of code
Python	3.9	Coding language
Python Libraries	Opencv.matplotlib	
GPS		

Hardware Requirements:

Hardware Device	Specification	Purpose
CPU	A PC with a descent graphic card	
RAM	8/12 GB	
Storage	128 GB	
Camera	High resolution cameras	To capture data

Table 3.2. Hardware Requirements

Raspberry Pi

Sensors and a Raspberry Pi comprise the hardware module. Every sensor node will be connected to the Raspberry Pi. When users request it, it provides the communication module with the most recent database status. A computer the size of a credit card is called the Raspberry

Pi. This is similar to an ordinary computer. The operating system that runs on the Raspberry Pi is called Raspbian, and it was based on Linux. The programming language used for it is called Python. The Raspberry Pi is connected to the users' Wi-Fi network using PHP. Connect the dongle to the Raspberry Pi in order to utilize Wi-Fi. Based on information gathered from parking lots, the overall movement of the automobile is controlled by the user-Raspberry Pi connection. Smart parking refers to the design, development, and production of cutting-edge parking technologies. It is a vehicle parking system that assists drivers in finding a vacant space. The Smart Parking system has been demonstrated to be an accurate, reliable, and cost-effective method of ensuring that users know exactly where unoccupied automobile parking places are.

3.3.3 Modules and its Description

In the project, there are 6 modules and their descriptions are given below:

Module 1. Parking Space Picker:

Creating a parking space picker using mouse events in OpenCV involves developing a graphical user interface that enables users to interactively select parking spaces by clicking on image regions. The first step in the procedure is to define a mouse callback function that registers the coordinates of clicked spots as parking spaces and listens for mouse events, specifically the click of the left button. The mouse callback function is designated to handle mouse events in the window that is formed when an image of the parking lot is loaded. When a user clicks on the image, visual markers, such green circles, are displayed at the clicked points' coordinates, which are then saved to signify the selected parking spaces. Users can hit a key to clear all specified areas in the software, and they can also press a different key to end the application. The intuitive selection of parking places is made possible by this interactive method.

Module 2. Image processing Layer:

Image preprocessing in OpenCV involves a set of techniques used to enhance the quality and suitability of images for further analysis or processing. These techniques encompass a range of operations aimed at improving image clarity, reducing noise, and extracting relevant features. Initially, images may undergo conversion to grayscale to simplify processing. Subsequently, smoothing and blurring filters such as Gaussian blur are applied to reduce noise and enhance edge detection. Thresholding techniques help in segmenting images by converting them into binary form based on pixel intensity values. Morphological operations like erosion and dilation further refine segmentation results by eliminating noise and filling gaps. Histogram equalization adjusts the contrast of images to improve their visual appearance, while scaling, rotation, and cropping operations enable standardization and alignment of images. Overall, image preprocessing in OpenCV plays a crucial role in optimizing images for various computer vision tasks, ultimately contributing to the accuracy and efficiency of downstream algorithms and analyses.

Module 3. Parking space detection:

An important module of an efficient parking management system is vacant space detection. There are multiple ways to detect these spaces without needing a large workforce. Modern technologies have allowed businesses to use convenient ways to manage their resources. Owners can use hardware like sensors to keep track of empty spaces in their parking facility. These sensors use ultrasonic and 3D vision technology to detect any empty parking space in the lot. This information is monitored by the system administrator and showcased to the vehicle owners. They can use this information to book the empty spaces beforehand. However, this system requires an individual sensor at each parking space and regular maintenance, which can be quite costly. Overall, image preparation in OpenCV is critical for optimizing pictures for diverse computer vision tasks, resulting in improved accuracy and efficiency of downstream algorithms and analysis.

Module 4. Shortest Path:

The Shortest Path module use an algorithm to locate the nearest available parking space. Because the parking lot's entrance is already specified, the algorithm begins at that point and represents the nearest available parking space.

Module 5. Vehicle detection:

There are several processes involved in preprocessing the input photos and identifying possible automobile zones in order to detect cars in OpenCV using blur, grayscale, and thresholding approaches. The input image is first transformed to grayscale in order to streamline the processing and lower the computational complexity. A blur filter, like Gaussian blur, is then used to improve edges and reduce noise, making it easier to differentiate automotive components. The image is then segmented into binary regions based on pixel intensity levels using thresholding, which successfully separates objects from the background. Contour detection methods are used to further process this binary image in order to find candidate automobile regions. Size, aspect ratio, and other attributes are used to identify contours that match to possible car forms.

Module 6. Integration of cameras:

Camera sensor integration is the period of time when the camera's clocks are set to trap and keep a charge. The behavior of the readout electronics acts as the integration's limit, and it is completely unrelated to the shutter's exposure. Size, aspect ratio, and other characteristics are utilized to determine shapes that are similar to potential automobile designs.

3.4 Requirements Engineering

There are two types of requirements for the module. They are functional and non-functional requirements.

3.4.1 Functional Requirements

Functional requirements for our project typically involve specific features and capabilities that the system must exhibit. Based on the description you provided, here are some potential functional requirements,

Parking Space Detection: Within the collected photos or video feed, parking spaces should be precisely detected and identified by the system. To find parking spaces, it ought to make use of object identification methods and computer vision techniques.

Real-time Updates: Users should get real-time notifications from the system on the occupancy status of parking spots, which should be constantly checked. This entails updating the availability of parking spaces when vehicles arrive and depart.

User Interface: Smartphones and other compatible devices should be able to quickly access and use the AR interface. Users should be able to move about the parking lot, zoom in and out, and pick particular spots to view more details.

Navigation Assistance: To direct customers to open parking spaces, the system ought to offer navigation support. In order to provide turn-by-turn directions to the designated parking spot, it ought to be integrated with GPS technology and mapping services.

Parking Space Reservation: Optionally, the system can include a feature that allows users to reserve parking spaces in advance. This would require integrating a reservation system that ensures the availability of the selected space upon arrival.

Data Analytics and Reporting: Optionally, the system can include data analytics capabilities to generate reports on parking utilization, occupancy rates, and other relevant metrics. This can provide valuable insights for parking management and optimization.

Security and Privacy: The system should ensure the security and privacy of user data. It should employ encryption techniques, secure data storage, and comply with relevant data protection regulations.

3.4.2 Non-Functional Requirements

Non-functional requirements for our project typically outline the criteria that the system should meet in terms of performance, security, and other quality aspects. Considering the nature of the project, here are some potential non-functional requirements,

Performance: For fast and precise parking spot detection, the system should be able to process and evaluate the recorded photos or video feed in real-time. It must be able to manage a large number of users while offering a flawless user experience free from noticeable latency or delays.

Reliability: The system need to be dependable and always accessible. It should operate continuously and make parking information available with little downtime, and it should be able to gracefully handle any system failures or disturbances.

Scalability: In order to handle an increasing number of individuals and parking spots, the system must be scalable. It should be built to accommodate future growth and integration with other parking lots while maintaining performance and user experience in the face of rising demand.

Compatibility: Many devices, such as tablets, smartphones, and other AR-capable gadgets, ought to work with the system. It should provide a consistent and ideal user experience across several platforms by supporting a range of operating systems and screen sizes.

Usability: The augmented reality interface must to be simple to use, intuitive, and easy to comprehend. It should require minimal training or technical knowledge for users to interact with the system effectively. To guarantee inclusion for people with impairments, the system should also take accessibility principles into account.

Accuracy: Parking spots should be accurately identified and classified using the parking space detection algorithm. It must to reduce false positives and false negatives while giving consumers accurate and dependable information regarding parking availability.

Security: User data security and privacy should be given top priority by the system. To shield user data from hostile attacks, illegal access, and data breaches, it should have strong security mechanisms in place. This covers adhering to privacy laws, using secure communication protocols, and encrypting sensitive data.

Maintenance and Support: There should be a clear maintenance and support plan for the system. To guarantee the system operates at peak efficiency and take care of any potential problems, it should include frequent updates, bug repairs, and additions. Sufficient technical assistance ought to be accessible to aid users with any questions or issues.

Integration: The system ought to be made to easily interface with other services or current parking management systems. It ought to provide interoperability and data interchange, enabling a thorough ecosystem for parking management.

Cost-effectiveness: It should be economical to install and maintain the system. It ought to take into account elements like the necessary hardware, software licenses, infrastructure costs, and continuing operating expenditures.

3.5 Analysis and Design through UML

Software system modelling and design are aided by visual diagrams known as UML (Unified Modelling Language) diagrams. There are different UML diagram kinds, each serving a particular purpose. These UML diagrams are some of the most common ones used in software development.

Class Diagram: Represents the static structure of the system, showcasing classes, attributes, methods, and their relationships.

Sequence Diagram: Visualizes the interactions and messages exchanged between objects or actors over time, showing the flow of events in a particular scenario or use case.

Use Case Diagram: Illustrates the interaction between actors (users or external systems) and the system, focusing on the system's functionalities from a user's perspective.

3.5.1 Class Diagram

Unified Modelling Language (UML), class UML diagrams represent the classes, attributes, methods, and connections inside a system. They function similarly to static structure diagrams. They provide a visual depiction of how classes behave and are organized in object-oriented systems. The class diagram explains the main components and their interactions in a smart parking space identification system that uses OpenCV. Detecting parking spaces, handling parking lot photos, and managing their statuses are all handled by the SmartParkingSystem class, which forms the basis of the system. Each parking space in the lot is represented by a Parking Space class, which is defined by its own identity, coordinates, and status (available or occupiedThe Smart Parking System class coordinates the finding of parking spaces by perhaps utilizing techniques like object recognition and contour detection in conjunction with OpenCV's image processing capabilities. Parking spots can have their statuses modified based on the presence of vehicles once they have been recognized. This class diagram gives developers the structural blueprint they need to create a smart parking system with OpenCV, allowing them to design and implement the system's main features.

Here are some major concepts and aspects of class UML diagrams

Class: Represents a template or blueprint for constructing objects. A class is illustrated as a rectangle with three compartments.

Attributes: represent the attributes or data associated with a class. Attributes describe the state or features of objects produced by the class.

Methods: They represent the actions or activities that objects in the class can do. Methods define the functionality or operations that objects can exhibit.

The Fig 3.2 Describes the Python implementation of the Smart Parking System class mentioned above, which uses the OpenCV library to identify and track parking spaces in a given video feed. The video source and parking spot coordinates are initialized first. The detect_cars method uses a Haar cascade classifier to recognize cars within the frame after the capture_frame method receives a frame from the video source. The positions of the detected automobiles are saved, and the analyze_parking_status method uses the overlap between the car and parking space regions to determine whether parking spots are occupied.Lastly, the display_parking_status method uses rectangles to graphically show the parking status. Red slots are occupied, while green spaces are empty. Until the user presses 'q' to end the session, the system keeps running in a loop and updates the parking status display in real-time. The basis of a smart parking system that can track parking space occupancy on its own is this class structure.

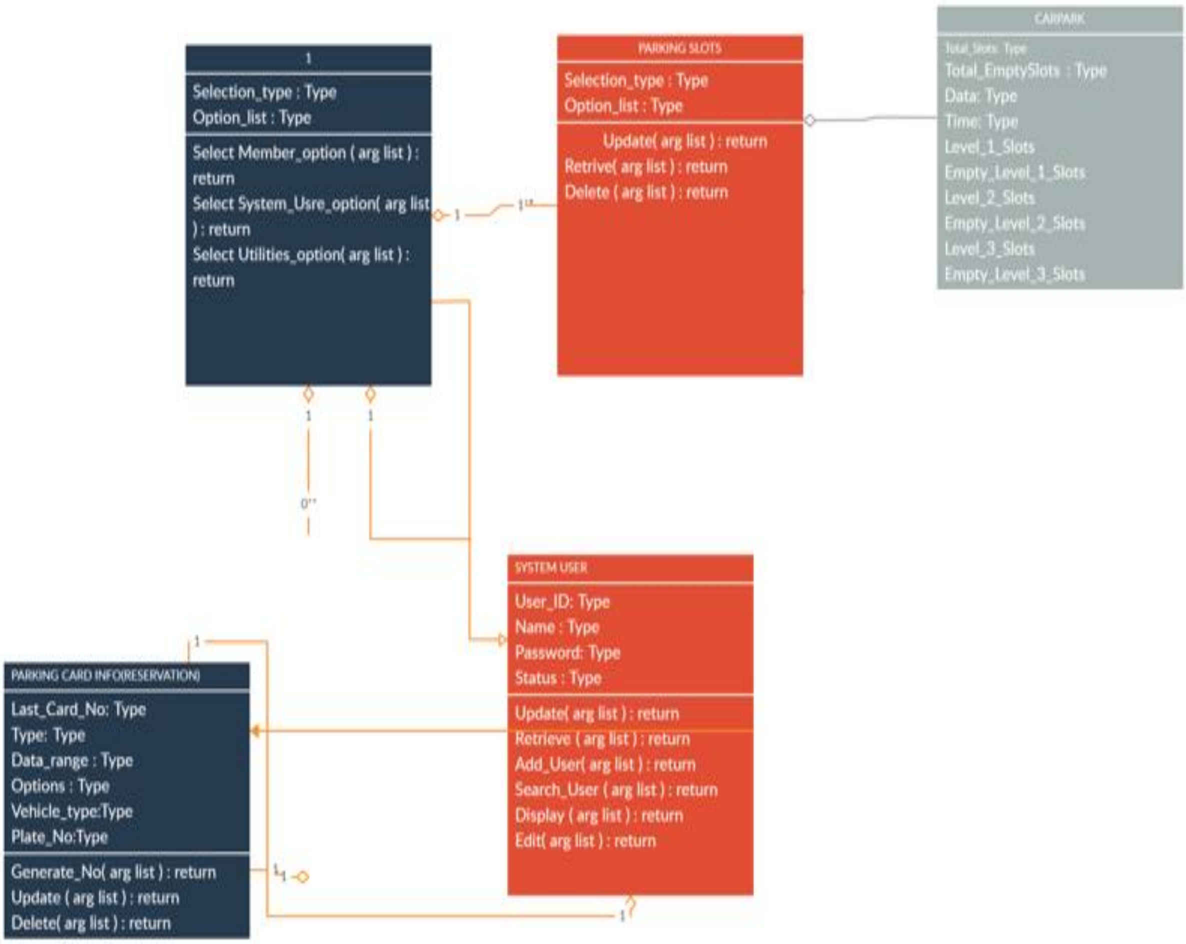

Figure 3.2 Class Diagram

3.5.2 Sequence Diagram

A series of An UML diagram is a particular sort of behavioral diagram in the Unified Modelling Language (UML) that depicts the communications and interactions that take place over time between actors or objects. By displaying the series of events or activities that take place in a certain scenario or use case, it illustrates the dynamic behavior of a system.

Key elements and concepts in sequence UML diagrams include:

Lifelines: Represent the participating objects or actors in the sequence diagram. Lifelines are depicted as vertical lines, usually labeled with the name of the object or actor they represent.

Messages: Represent the communication or interaction between lifelines. Messages are shown as arrows or dashed lines with labels indicating the nature of the communication. They can be synchronous (denoted by a solid line), asynchronous (denoted by a dashed line), or self-referential (loopback arrow).

Return Messages: Indicate the response or return value from a method call. Return messages are depicted as dashed arrows, often labeled with the return value or the message "return".

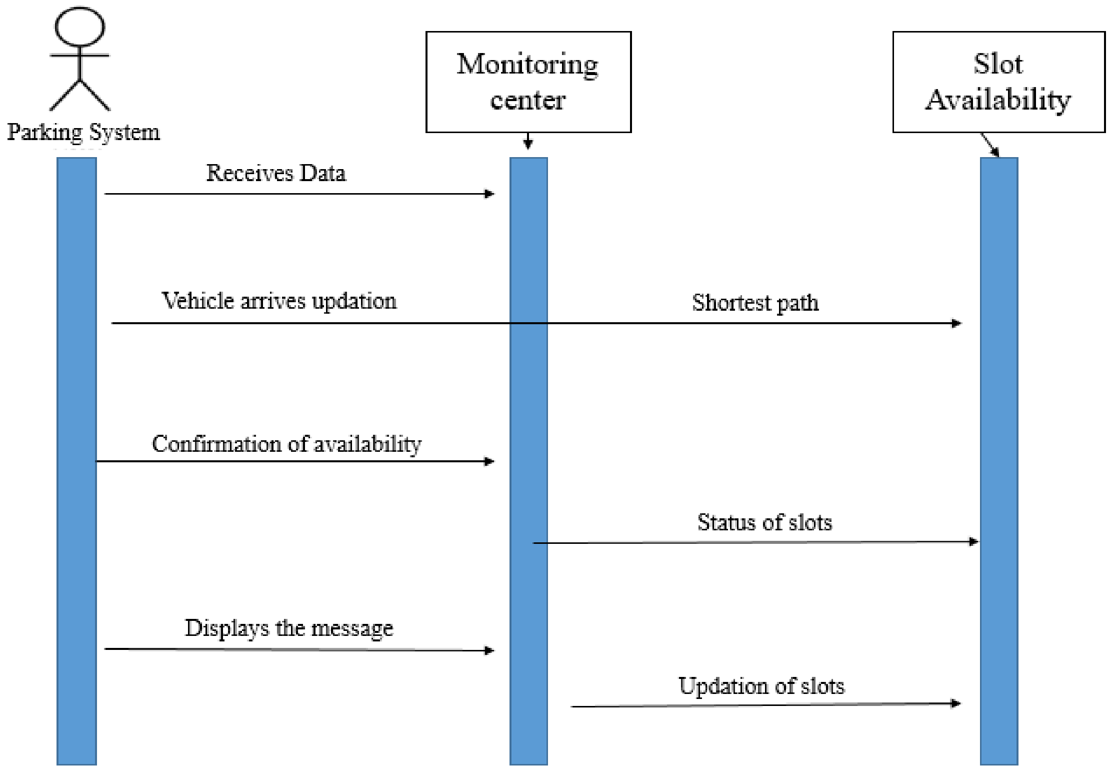

Figure 3.3 Sequence Diagram

3.5.3 Use Case Diagram

Unified Modelling Language (UML), a UML diagram is a behavioral diagram that shows how actors, or humans or external systems, interact with a system. It focuses on presenting the many use cases and how actors interact with the system to accomplish particular objectives, all while capturing the functionality of the system from the viewpoint of the user.

Key elements and concepts in use case UML diagrams include

Actors: Represent the roles or entities outside the system that interact with it. Actors might be people, other systems, or even hardware components. They are depicted as stick figures or marked blocks.

Use Cases: Represent the specific functionalities or tasks that the system provides to its users. Use cases describe the interactions between actors and the system to accomplish a specific goal. They are depicted as ovals or ellipses and labelled with descriptive names.

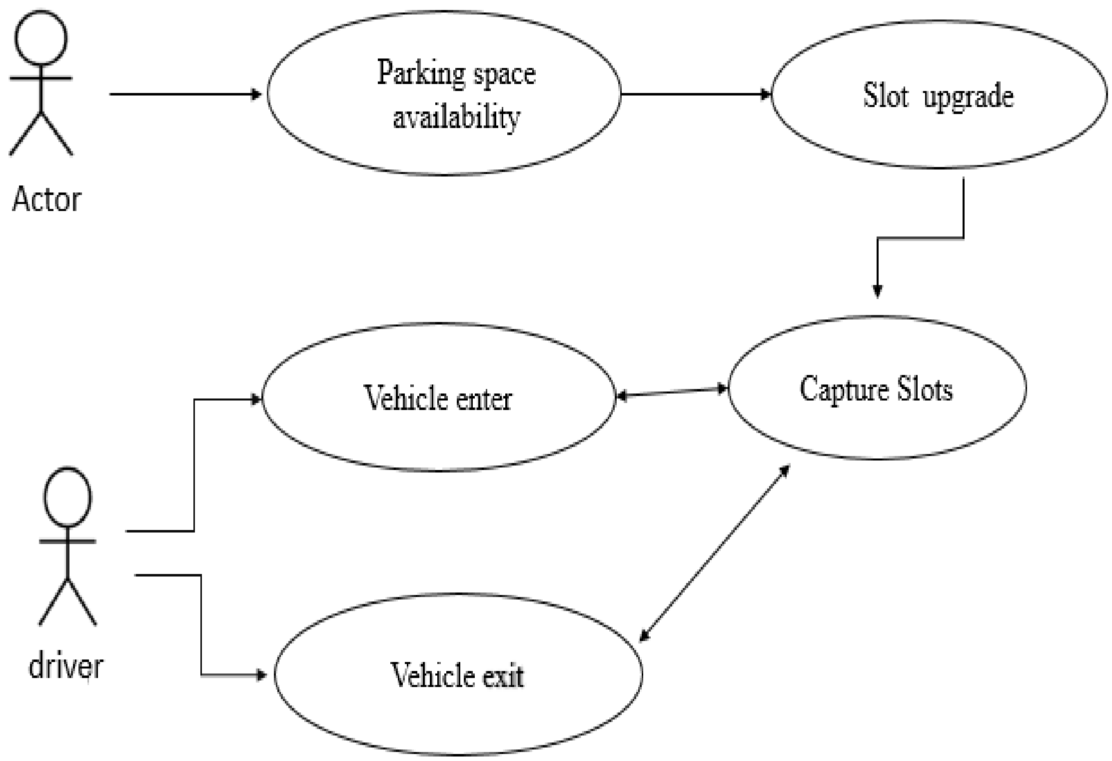

Figure 3.4 Use Case Diagram

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results

Firstly**,** parking space picker in OpenCV allows users to define parking slots by mouse click events in a graphical user interface. Users can click on an image representing a parking lot to designate the boundaries of individual parking spaces. Each mouse click event captures the coordinates of the clicked points, which are then used to define rectangular regions corresponding to parking slots. These regions are typically stored as data points or coordinates for further processing. The parking space picker provides a user-friendly way to create and customize parking layouts, facilitating tasks such as parking occupancy detection and management in smart parking systems as shown in Fig 4.1.

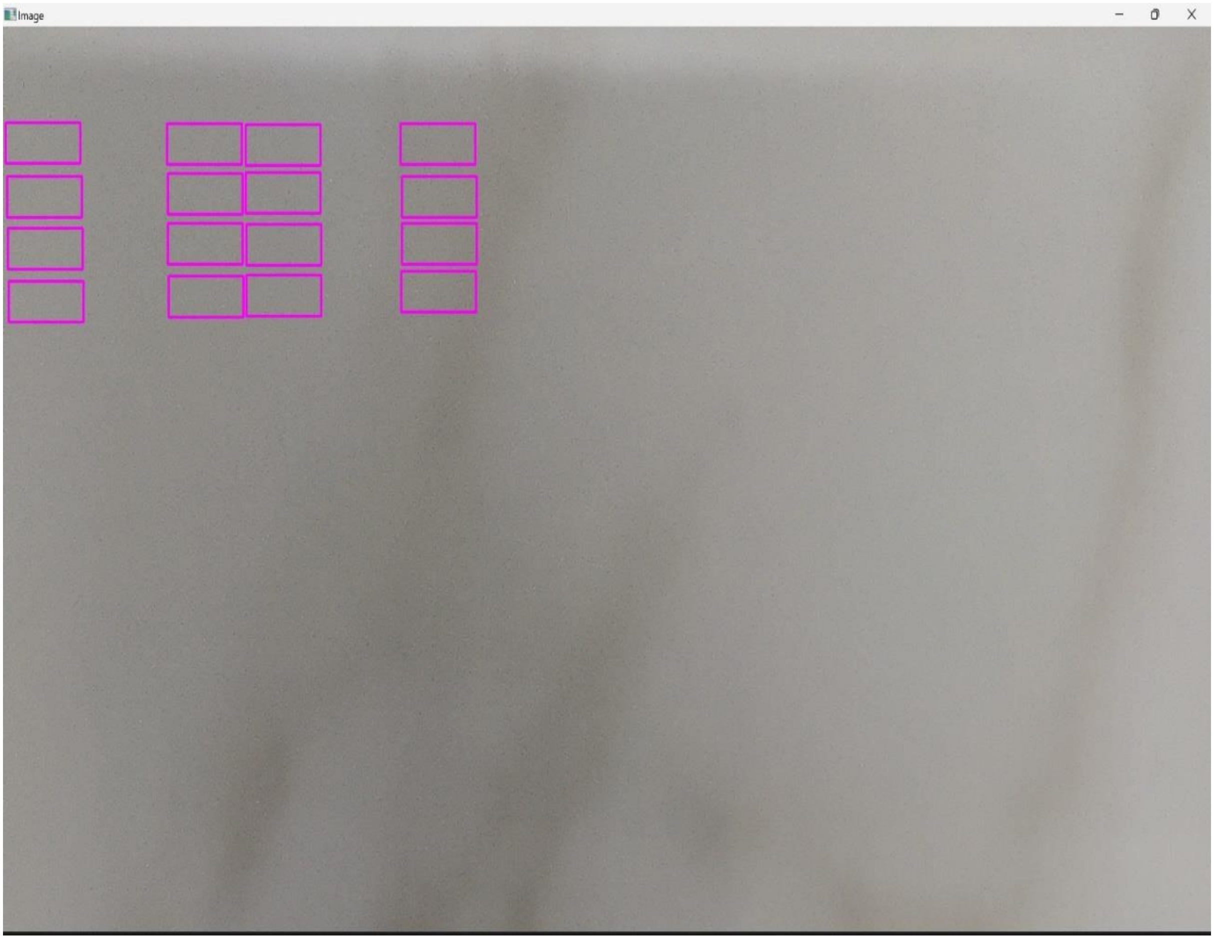

Figure 4.1 Parking Space Picker

Thresholding involves setting pixel values in an image above or below a certain threshold to specific values. The input image, the threshold value, the maximum pixel value, and the thresholding technique are among the parameters required by this function. The function produces a binary picture after performing the thresholding procedure, with pixels set to either 0 or the maximum pixel value depending on whether they satisfy the given threshold conditions. In computer vision applications, thresholding is frequently used for tasks including object detection, feature extraction, and image segmentation.as shown in Fig 4.2.

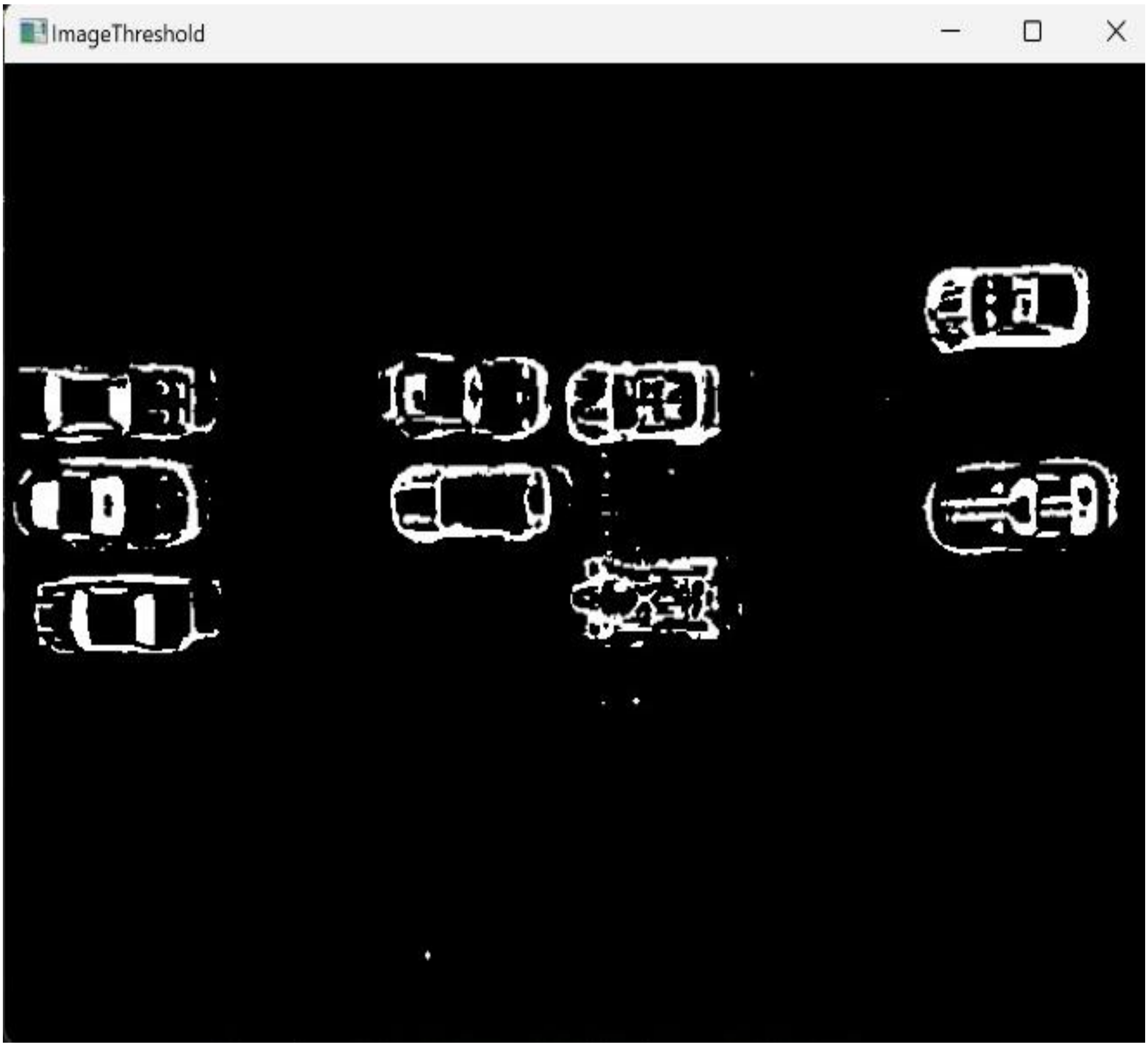

Figure 4.2 Image Threshold preprocessing

The median blur operation in OpenCV is a type of spatial filtering technique used to reduce noise in an image while preserving edges and other important image features. In contrast to conventional blurring techniques, which substitute the nearby pixel average for the center pixel value, median blur substitutes the neighboring pixel value for the central pixel value. Because of this, median blur works especially well at eliminating "salt-and-pepper noise," which is the effect of individual pixels in an image that differ noticeably from their surrounds. Outlier pixel values are less significant in the final image when the median value is used, producing a smoother, edge-preserved image as shown in Fig 4.3.

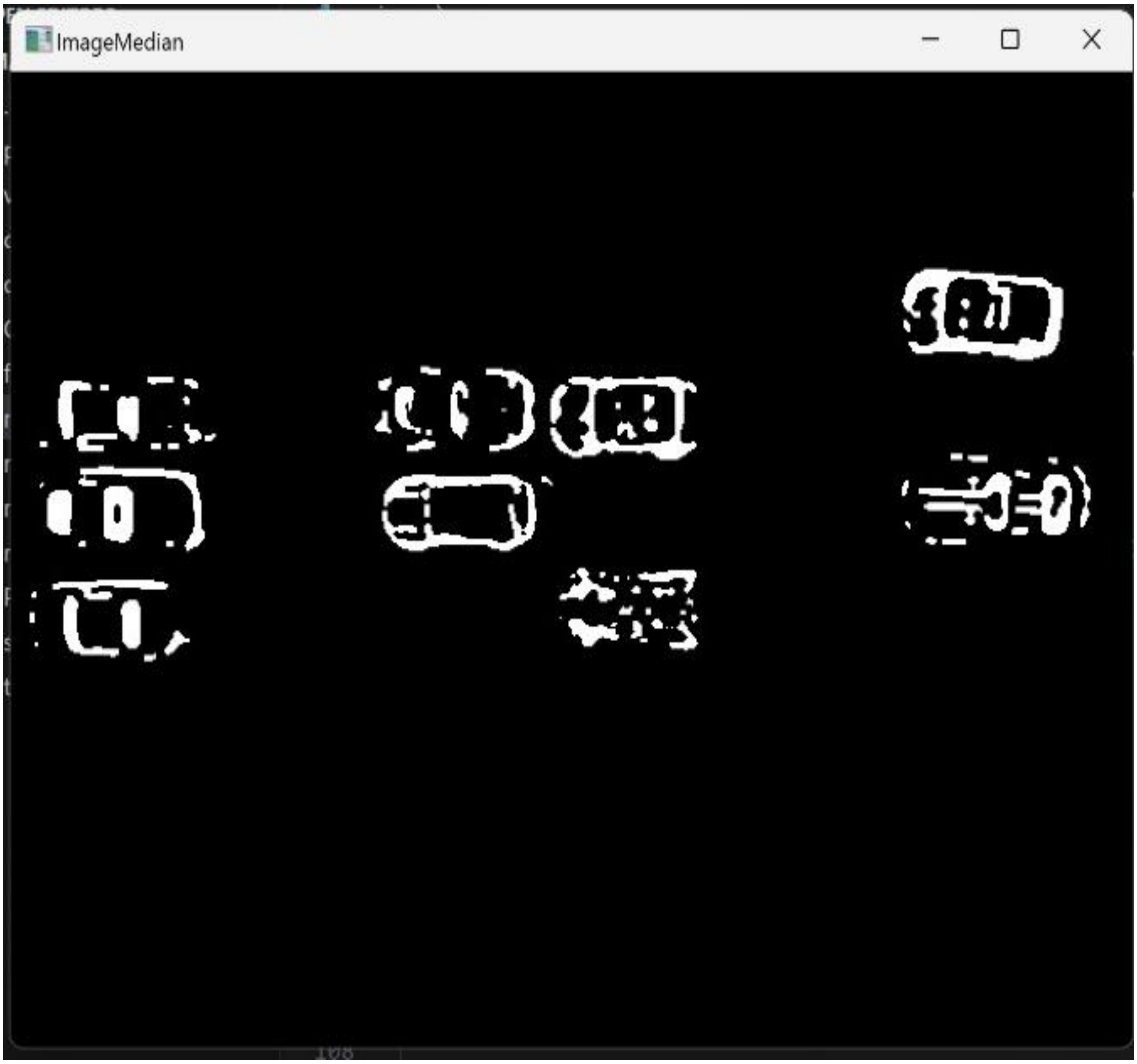

Figure 4.3 Image Median Blur preprocessing

The dilation operation in image processing, particularly in OpenCV, is a morphological operation used to expand regions of foreground pixels in an image. The process entails moving a structuring element—usually a square or rectangular kernel—over the image and substituting the maximum pixel value discovered in the kernel's vicinity for the central pixel in the kernel. This technique successfully fills up tiny gaps or holes in the items inside the image and enlarges bright areas. In many computer vision applications, dilation is combined with other morphological procedures like erosion to accomplish objectives like object segmentation, edge identification, and noise reduction as shown in Fig 4.4.

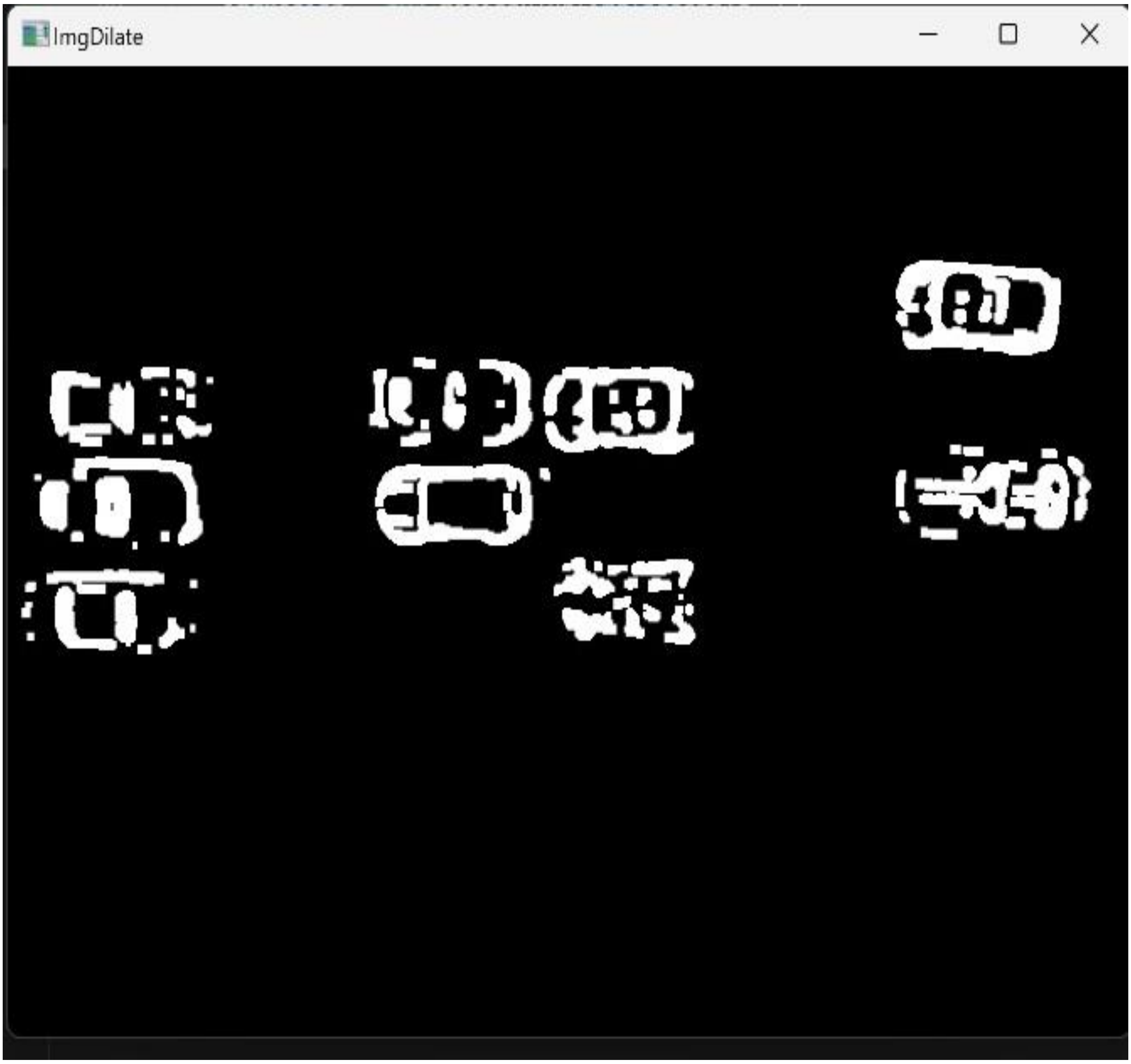

Figure 4.4 Image Dilate preprocessing

Converting an image from BGR to grayscale in OpenCV involves a straightforward process. Initially, the image is loaded into memory using the cv2.imread () function. Then, the conversion from BGR to grayscale is executed, resulting in a grayscale representation of the original image. The converted grayscale image can be further processed or displayed using OpenCV's functionalities as required. This process encapsulates the essential steps of loading an image and performing a color space conversion, contributing to various computer vision tasks and applications as shown in Fig 4.5.

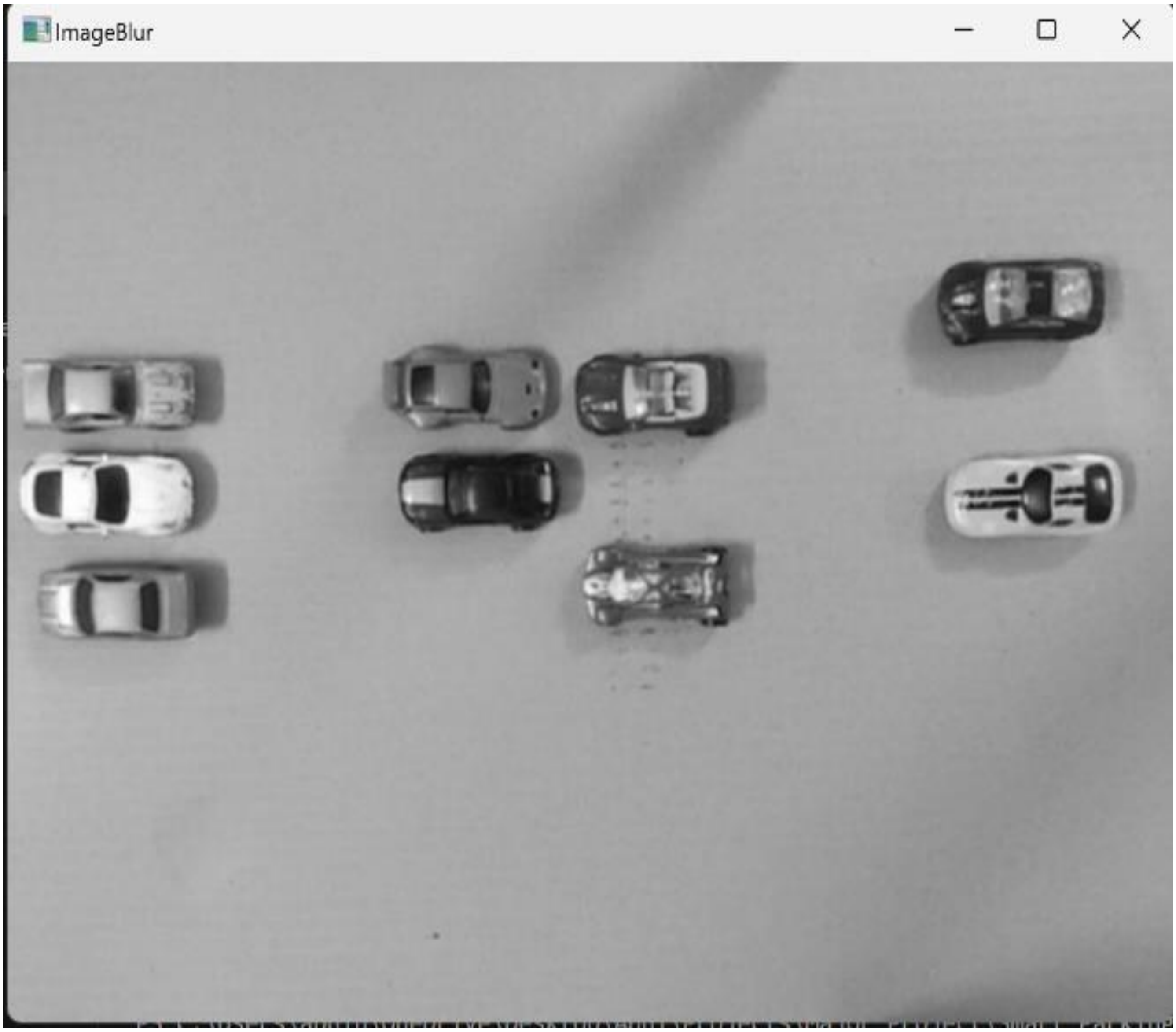

Figure 4.5 Image Blur preprocessing

A smart parking system implemented with OpenCV typically involves several key steps. The technology first records live video from cameras positioned in parking lots. Vehicles in the video stream are recognized and tracked by the system using computer vision techniques like object detection. The system uses occupancy status and vehicle positions to decide which parking spaces are available. Users can effectively locate and go to empty parking spaces thanks to the processing and presentation of this data via a user interface. The system can also produce reports and statistics on parking utilization, which will help with resource optimization and management. All things considered, the smart parking system makes use of OpenCV's image processing and analysis capabilities to improve parking management effectiveness and deliver real-time parking occupancy statistics as shown in Fig 4.6 and Fig 4.7.

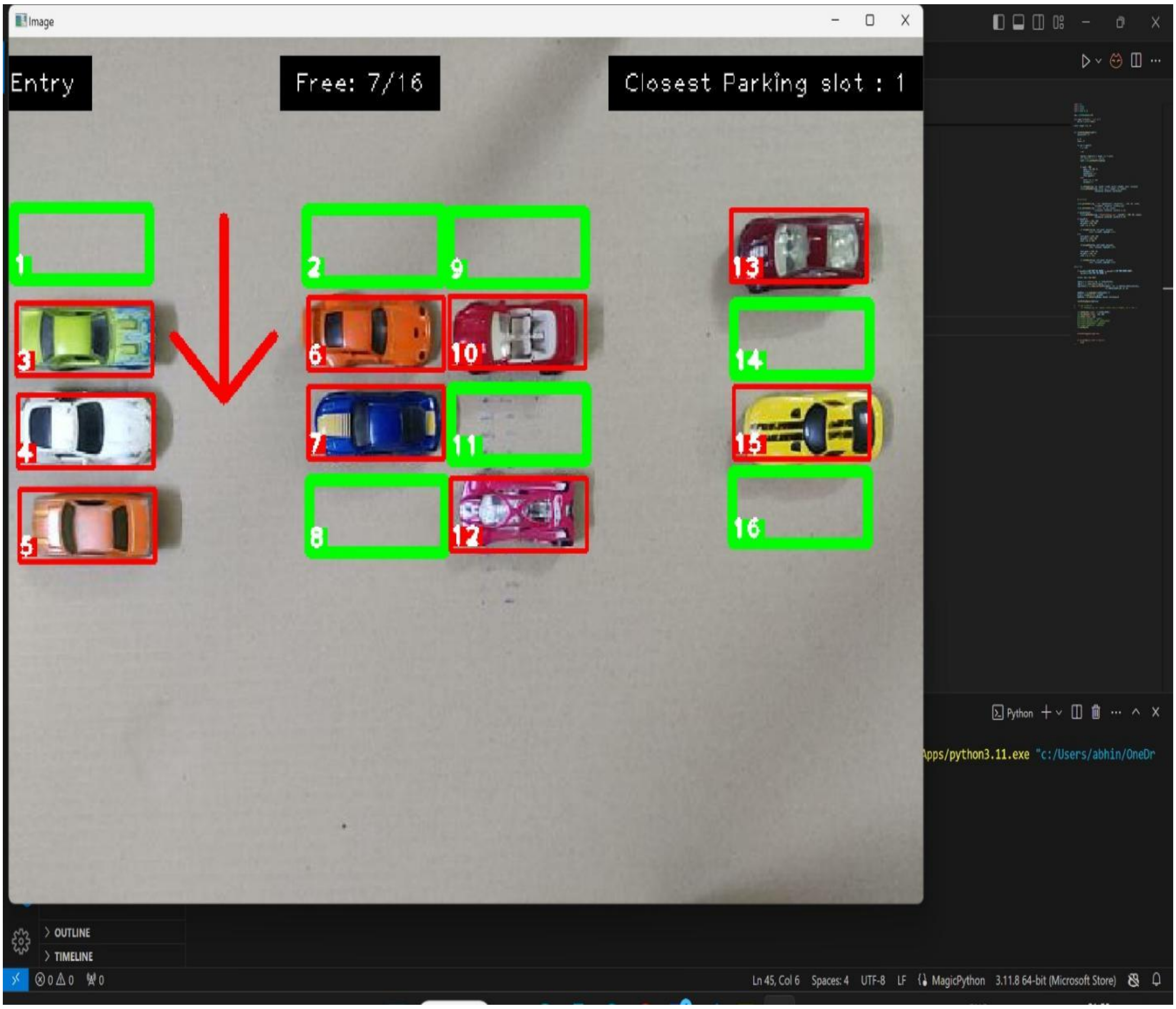

Figure 4.6 Representation of parking occupancy

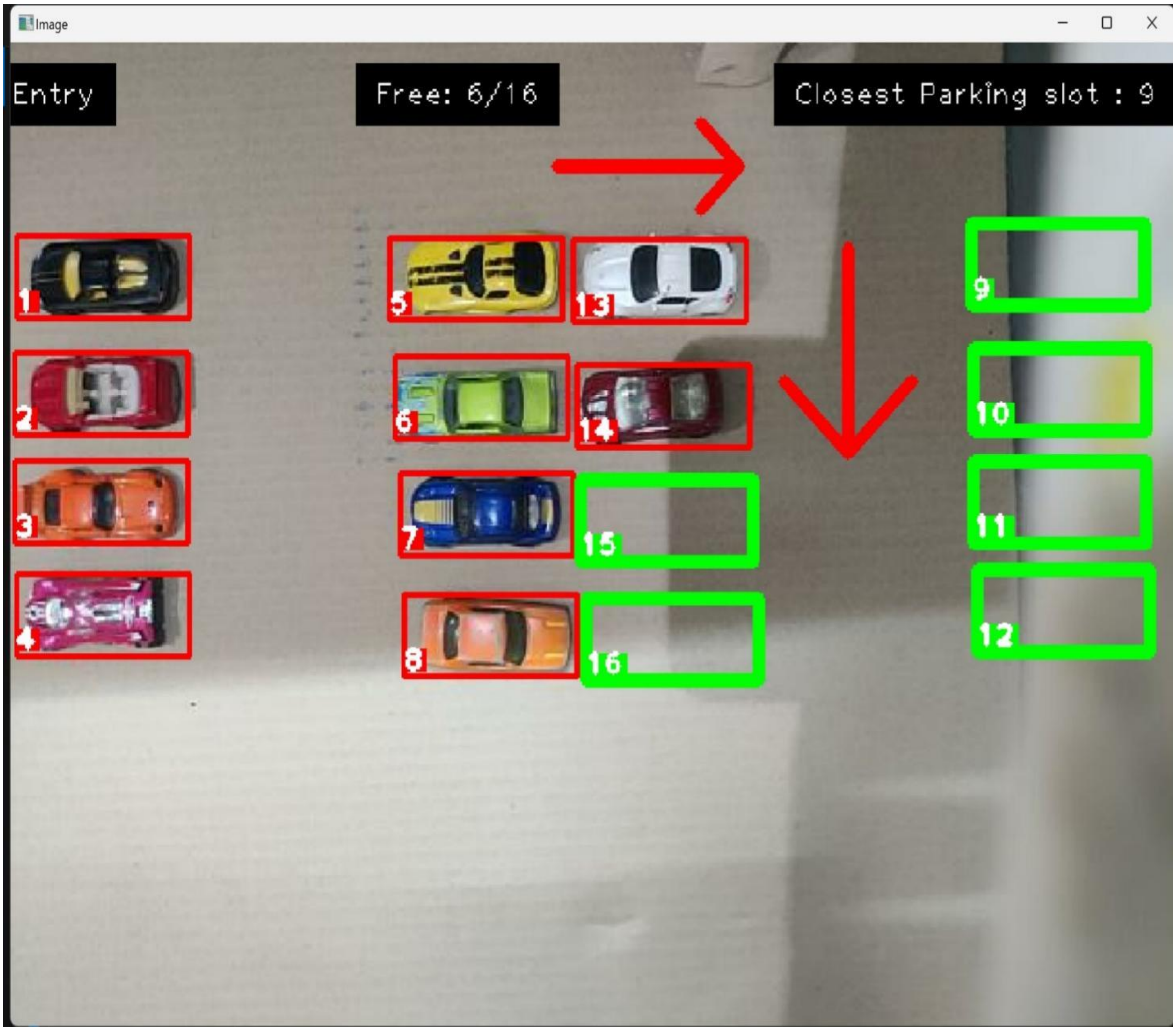

Figure 4.7 Representation of Parking Slot with Navigation

4.2 Discussion

The smart parking system, leveraging OpenCV technology, has achieved an impressive accuracy rate of 86%. This accuracy denotes the system's capability to effectively differentiate between occupied and vacant parking spaces. Compared to traditional methods, such as manual monitoring or sensor-based systems, the OpenCV-based approach offers several advantages. Firstly, it provides a cost-effective solution as it relies primarily on software algorithms rather than expensive hardware installations. Secondly, the flexibility of OpenCV allows for easy customization and adaptation to different parking lot layouts and environmental conditions. Additionally, the real-time processing capability of OpenCV ensures swift and accurate detection, enabling efficient management of parking resources. In contrast, manual monitoring methods are prone to human error and are often labor-intensive, while sensor-based systems may incur high installation and maintenance costs. Thus, the utilization of OpenCV in the smart parking system not only enhances accuracy but also offers a more scalable, cost-effective, and adaptable solution for efficient parking management.

Assessing the accuracy of a smart parking system involves quantifying its ability to correctly detect and classify parking space occupancy. This entails measuring metrics such as precision, recall, and F1 score, which provide insights into the system's performance. For instance, a precision of 90% indicates that 90% of the detected occupied parking spaces are indeed occupied, while a recall of 85% signifies that 85% of the actual occupied spaces are correctly identified by the system. Similarly, the F1 score, which balances precision and recall, offers a single metric to evaluate overall performance. In practical terms, achieving a precision of 90% means that out of 100 detected occupied parking spaces, 90 are correct, while a recall of 85% implies that out of 100 actual occupied spaces, 85 are successfully detected. These numerical values provide tangible insights into the system's accuracy and effectiveness in monitoring parking space occupancy, enabling stakeholders to make informed decisions regarding system optimization and deployment.

Evaluating the accuracy of a smart parking system often involves comparing multiple models or algorithms, each utilizing different approaches for car detection and parking space occupancy classification. For instance, a system may employ traditional computer vision techniques such as Haar cascades or HOG (Histogram of Oriented Gradients) combined with SVM (Support Vector Machine) classifiers, as well as modern deep learning-based methods like Faster R-CNN or YOLO (You Only Look Once). Each model may achieve varying levels of accuracy based on factors such as complexity, training data quality, and computational requirements.For example, let's consider three models: Model A utilizes Haar cascades for car detection and simple thresholding for parking space occupancy classification, Model B employs a Faster R-CNN architecture for both car detection and parking space classification, and Model C combines a pre-trained deep learning model for car detection with a custom SVM classifier for parking space occupancy determination. Through rigorous testing and validation, Model A achieves a precision of 85%, recall of 80%, and F1 score of 82%, Model B attains a precision of 92%, recall of 88%, and F1 score of 90%, while Model C demonstrates a precision of 88%, recall of 85%, and F1 score of 86%.These numerical values offer insights into the performance of each model, enabling stakeholders to make informed decisions regarding the selection and optimization of the smart parking system. Moreover, by comparing the accuracy of different models, developers can identify strengths and weaknesses, refine algorithms, and ultimately enhance the overall effectiveness of the system in accurately monitoring parking space occupancy.

As a result, the application of OpenCV technology highlights its importance in the field of smart parking management systems by improving accuracy and providing a scalable, affordable, and flexible solution for optimizing parking operations. Studies on the accuracy of parking systems have shown that different approaches perform differently. Accuracy rates for smart parking systems employing a variety of technologies, including computer vision, machine learning, and sensor networks, have been claimed in a number of published articles to be between 70% and 95%. While some studies have achieved high accuracy through sophisticated algorithms and sensor fusion techniques, others have encountered challenges related to environmental factors, occlusions, and dataset variability, leading to lower accuracy levels. Overall, advancements in technology and algorithm development have contributed to improving the accuracy of parking systems, yet ongoing research aims to address remaining challenges and further enhance performance in real-world deployments. A parking system utilizing machine learning (ML) techniques has been developed to accurately monitor parking space occupancy in real-time. By employing ML algorithms such as support vector machines (SVM) and convolutional neural networks (CNN), the system is capable of analyzing video streams or sensor data to classify parking spaces as either occupied or vacant. The ML models are trained on a labeled dataset comprising images or sensor readings of parking spaces under various conditions, allowing them to learn patterns associated with both occupied and vacant spaces. Through iterative training and optimization, the system achieves high accuracy in detecting parking space occupancy, often exceeding 90%. Additionally, the ML-based approach offers scalability and adaptability, enabling deployment across diverse parking environments. Furthermore, ongoing research focuses on enhancing the system's performance through techniques such as transfer learning, data augmentation, and model compression, with the ultimate goal of creating an efficient and reliable parking management solution.

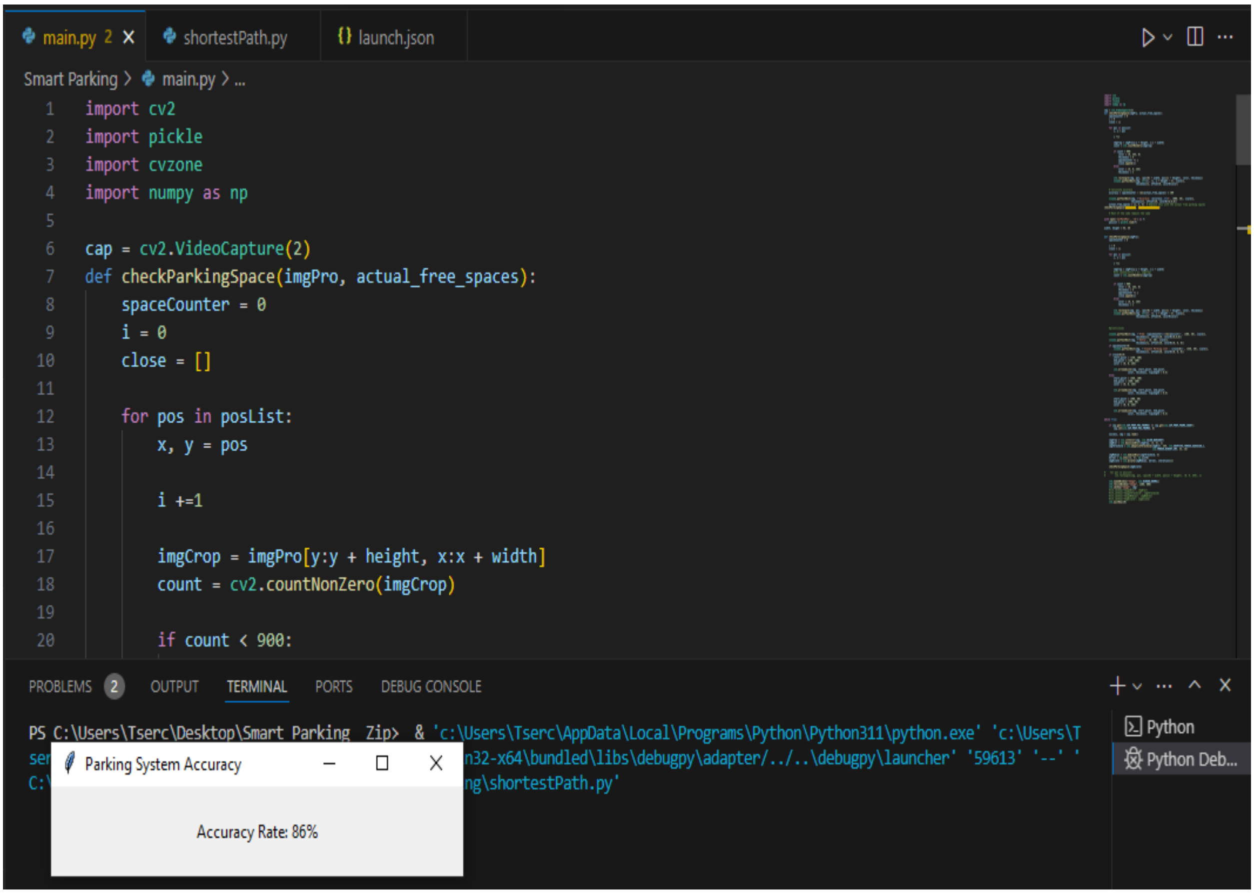

Figure 4.8 Accuracy

Fig 4.8 Describes smart parking system, leveraging OpenCV technology, has achieved a remarkable accuracy rate of 86%, a performance that stands out in comparison to other parking management systems documented in the literature. Traditional approaches such as manual monitoring or sensor-based systems are prevalent but often exhibit limitations. Manual monitoring, for instance, relies heavily on human observation and is prone to errors, leading to inaccuracies in occupancy detection. On the other hand, while sensor-based systems offer automation, they are often cost-prohibitive due to the need for extensive hardware installations and ongoing maintenance. In contrast, the OpenCV-based approach provides several notable advantages. Firstly, it offers a cost-effective solution by primarily relying on software algorithms, thereby minimizing the need for expensive hardware infrastructure. This cost efficiency makes it accessible to a wider range of parking facilities, including smaller businesses and public parking areas. Secondly, the adaptability of OpenCV allows for seamless customization to diverse parking lot layouts and environmental conditions. This flexibility ensures that the system can be easily tailored to suit specific requirements, such as different parking space dimensions or varying lighting conditions throughout the day. Moreover, the real-time processing capabilities of OpenCV enable swift and accurate detection of parking space occupancy, contributing to more efficient parking resource management. This real-time monitoring ensures that parking availability information is continuously updated, enabling drivers to quickly locate vacant spaces and reducing congestion within parking facilities. Additionally, the scalability of OpenCV allows for easy expansion or integration with existing systems, making it a versatile solution for both small-scale parking lots and large multi-level parking structures. In summary, the adoption of OpenCV technology in smart parking systems represents a significant advancement in parking management. Its high accuracy, cost-effectiveness, adaptability, and real-time processing capabilities make it a compelling alternative to traditional parking systems, offering improved efficiency and convenience for both parking facility operators and drivers alike.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Conclusion

This paper contributes to the growing field of Location-Based Services and showcases the potential of Augmented Reality in transforming driving navigation and parking assistance. With the use of Opencv, sophisticated navigation systems, and machine learning, the proposed strategy offers a workable means of enhancing urban mobility and increasing the enjoyment and sustainability of driving. In addition, it addresses the challenge of finding parking spots in urban areas and saves vehicles time and energy by eliminating the need for block circles. Experiments evaluating the proposed system have produced informative outcomes. While the system typically does a good job of placing visual cues appropriately and offering interactive driving instruction, reliability and stability issues exist, especially with regard to localization and object placement. The shortcomings of VPS technology in high-speed driving situations and on highways and roads underscore the necessity for more advancements. Notwithstanding these difficulties, the system exhibits a lot of promise as a substitute for conventional navigation techniques. When it works well, it provides a fun and immersive augmented reality navigation experience that can improve customer satisfaction and lessen annoyance when parking and driving.

5.2 Future Enhancements

Future research should focus on Subsequent investigations ought to concentrate on resolving the concerns of stability and reliability by either enhancing the functionality of VPS technology or including supplementary localization techniques, especially for situations involving fast driving. Furthermore, a lane detector that precisely locates the driver's present lane's boundaries can greatly enhance the navigator's readability.

Improved Accuracy and Reliability: Future developments may focus on enhancing the accuracy of AR-based parking space detection through more sophisticated machine learning algorithms and sensor fusion techniques. This could reduce false positives and negatives in identifying available spots.

3D Mapping and Object Recognition: Advancements in AR technology may involve 3D mapping of parking areas, allowing for better object recognition and spatial mapping of available parking spaces. This could include detecting obstacles or irregularities within spaces.

Enhanced User Interfaces: Future systems might introduce more intuitive and customizable AR interfaces for drivers. This could involve personalized preferences, such as preferred parking spot sizes or locations, and streamlined navigation within parking lots.

Integration with Autonomous Vehicles: As autonomous vehicle technology advances, AR-enabled parking systems could seamlessly integrate with these vehicles. AR could guide the vehicles to designated parking spots autonomously, reducing the need for human intervention.

Dynamic Traffic Flow Management: AR-equipped smart parking systems may evolve to offer insights into overall traffic flow within parking lots. This could assist in dynamically managing traffic by redirecting vehicles to less congested areas or optimizing entry and exit routes.

Multi-Sensor Fusion: In the future, systems may integrate AR with other sensors (such LiDAR, radar, or sophisticated cameras) to provide users a more complete picture of the parking area. This combination might improve dependability and accuracy.

Integration with Smart City Initiatives: AR-based parking systems could align with broader smart city initiatives, allowing data collected from parking spaces to contribute to urban planning, congestion management, and sustainability efforts.

Expanded Accessibility: Future developments may focus on making AR-enabled parking systems more accessible across different platforms and devices, ensuring compatibility with a wide range of smartphones, wearable tech, and vehicles.

APPENDICES

Sample Code

Main.py

```
import cv2

import pickle

import cvzone

import numpy as np

cap = cv2.VideoCapture(2)

with open('CarParkPos', 'rb') as f:

    posList = pickle.load(f)

width, height = 140, 80

def checkParkingSpace(imgPro):

    spaceCounter = 0

    i = 0

    close = []

    for pos in posList:

        x, y = pos

        i +=1

        imgCrop = imgPro[y:y + height, x:x + width]

        # cv2.imshow(str(x * y), imgCrop)

        count = cv2.countNonZero(imgCrop)

        if count < 900:

            color = (0, 255, 0)

            thickness = 5

            spaceCounter += 1

            close.append(i)

        else:

            color = (0, 0, 255)
```

```
            thickness = 2
        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), color, thickness)
        cvzone.putTextRect(img, str(i), (x, y + height - 3), scale=1,
                           thickness=2, offset=0, colorR=color)
    print(close)
    cvzone.putTextRect(img, f'Free: {spaceCounter}/{len(posList)}', (0, 30), scale=1,
                           thickness=1, offset=10, colorR=(0,0,0))
    cvzone.putTextRect(img, f'Entry', (280, 30), scale=1,
                           thickness=1, offset=10, colorR=(0, 0, 0))
    if spaceCounter>0:
        cvzone.putTextRect(img, f'Closest Parking slot : {close[0]}', (490, 30), scale=1,
                           thickness=1, offset=10, colorR=(0, 0, 0))
while True:
    if cap.get(cv2.CAP_PROP_POS_FRAMES) ==
cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
    success, img = cap.read()
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)
    imgThreshold = cv2.adaptiveThreshold(imgBlur, 255,
cv2.adaptive_thresh_gaussian_c,
                                         cv2.thresh_binary_inv, 25, 16)
    imgMedian = cv2.medianBlur(imgThreshold, 5)
    kernel = np.ones((3, 3), np.uint8)
    imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)
    checkParkingSpace(imgDilate)
#    for pos in posList:
#        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), (0, 0, 255), 1)
    cv2.imshow("Image", img)
```

```
    #cv2.imshow("ImageBlur", imgBlur)
    #cv2.imshow("ImageThreshold", imgThreshold)
    #cv2.imshow("ImageMedian", imgMedian)
    #cv2.imshow("ImgDilate", imgDilate)
    cv2.waitKey(10)
'''
    #checkParkingSpace(imgDilate)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
'''
```

Parking Space Picker

```
import cv2
import pickle
width, height = 140, 80
try:
    with open('CarParkPos', 'rb') as f:
        posList = pickle.load(f)
except:
    posList = []
def mouseClick(events, x, y, flags, params):
    if events == cv2.EVENT_LBUTTONDOWN:
        posList.append((x, y))
    if events == cv2.EVENT_RBUTTONDOWN:
        for i, pos in enumerate(posList):
            x1, y1 = pos
            if x1 < x < x1 + width and y1 < y < y1 + height:
```

```python
            posList.pop(i)
    with open('CarParkPos', 'wb') as f:
        pickle.dump(posList, f)
while True:
    img = cv2.imread('carParkImg.png')
    for pos in posList:
        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), (255, 0, 255), 2)
    cv2.imshow("Image", img)
    cv2.setMouseCallback("Image", mouseClick)
    cv2.waitKey(1)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
```

REFERENCES

[1] https://cybersnowden.com/smart-parking-smart-parking-with-iot-2023/

[2] https://www.javatpoint.com/opencv

[3] Gaurav Satyanath, Jajati Keshari Sahoo, Rajendra Kumar Roul "Smart Parking Space Detection under Hazy conditions using Convolutional Neural Networks: A NovelApproach"

[4] Rifath Mahmud, A.F.M Saifuddin Saif, Dipta Gomes "A Comprehensive Study of Real-TimeVacant Parking Space Detection Towards the need of a Robust Model" AIUB JOURNAL OF SCIENCE AND ENGINEERING ISSN: 1608 - 3679 ,pg no:2520 - 4890

[5] Purva Hattale, Vikrant Jangam, Shrutika Khilare, Yash Ratnaparkhi, Pradnya Kasture "Parking Space Detection Using Image Processing" International Journal of Science and Research (IJSR) ISSN: 2319-7064 SJIF (2019): 7.583

[6] Xiangwu Ding , Ruidi Yang "Vehicle and Parking Space Detection Based on Improved YOLO Network Model" Journal of Physics: Conference Series 1325 (2019) 012084,doi:10.1088/1742-6596/1325/1/012084

[7] ElakyaR,Juhi Seth, Pola Ashritha, R Namith "Smart Parking System using IoT"International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online),Volume-9 Issue-1, October, 2019

[8] Xumin Huang, Peichun Li, "FedParking: A Federated Learning based Parking Space Estimation with Parked Vehicle assisted Edge Computing"2021,https://doi.org/10.48550/arXiv.2110.128

[9] Abhirup Khanna, Rishi Anand "T-YOLO: TinyVehicle DetectionBased on YOLOand Multi-ScaleConvolutionalNeural Networks" International Conference on Internet of Things and Applications (IOTA),DOI: 10.1109/IOTA.2016.7562735

[10] Liangliang Lou, Qiang Li, Zengchao Zhang, Rong Yang, Wei He "An IoT-Driven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System" IEEE Internet of Things Journal DOI: 10.1109/JIOT.2020.2992431

[11] Arun Varghese; G. Sreelekha “An Efficient Algorithm for Detection of Vacant Spaces in Delimited and Non-Delimited Parking Lots” IEEE Transactions on Intelligent Transportation Systems DOI: 10.1109/TITS.2019.2934574

[12] Zhanghua Cai, Yantao Zhou, Zibin Weng, Lei Deng, Yunlong Luo, Ming Yu, Yihong Qi “3-D Printing Conformal K-Band Lens Antenna for a Smart Parking Space Detection System” IEEE Transactions on Instrumentation and Measurement DOI: 10.1109/TIM.2020.2988137

[13] Yuanfeng Wu, Xinran Guan, Boya Zhao, Li Ni, Min Huang “Vehicle Detection Based on Adaptive Multimodal Feature Fusion and Cross-Modal Vehicle Index Using RGB-T Images” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing DOI: 10.1109/JSTARS.2023.3294624

[14] Sai Nikhil Reddy Mettupally, Vineetha Menon “A Smart Eco-System for Parking Detection Using Deep Learning and Big Data Analytics” DOI: 10.1109/SoutheastCon42311.2019.9020502 INSPEC Accession Number: 19433821

[15] Chin-Kit Ng, Soon-Nyan Cheong, Erfan Haji Mohammad hosseinmemar, Wen-Jiun Yap “Mobile outdoor parking space detection application” 2017 IEEE 8th Control and System Graduate Research Colloquium DOI: 10.1109/ICSGRC.2017.8070573 INSPEC Accession Number: 17262302

[16] Sherzod Nurullayev,Lee, “Generalized Parking Occupancy Analysis Based on Dilated Convolutional Neural Network”, Sensors 2019, vol:19(2), pgno:277-350,https://doi.org/10.3390/s19020277

[17] Vladimir, Josef, S. Ou, M. de Heaver, " A Smart Parking System Based on Mini PC Platform and Mobile Application for Parking Space Detection," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 2234-2245, 2018.

[18] A. Khanna, R. Anand, "IoT based smart parking system," Proceedings of International Conference on Internet of Things and Applications (IOTA), pp. 266-270, Pune, India, January 2016.

[19] Danakorn Nincarean Eh Phon, Mohamad Bilal Ali, “Collaborative Augmented Reality in Education: A Review”, 2014 International Conference.2014, https://doi.org/10.1109/LaTiCE.2014.23

[20] Jaspreet Singh; Urvashi “Augmented Reality Technology: Current Applications, Challenges and itsFuture”20224th International Conference, 2022,

https://doi.org/10.1109/ICIRCA54612.2022.9985665.

[21] Xiang Li, Yuan Tian, “Object Detection in the Context of Mobile Augmented Reality”, 2020 ,IEEE International Symposium on Mixed and Augmented Reality (ISMAR),2020ISSN:1554-7868, https://doi.org/10.1109/ISMAR50242.2020.0035

[22] Yanlei Gu,Woranipit Chidsin., “AR-based Navigation Using Hybrid Map”, 2021 IEEE 3rdGlobal Conference on Life Sciences and Technologies (LifeTech),2021, https://doi.org/10.1109/LifeTech52111.2021.9391866.

[23] Hans Chandra,Michael, “Smart Parking Management System: An integration of RFID, ALPR, and WSN” 2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS),2018, https://doi.org/10.1109/ICETSS.2017.8324174.

[24] Madhur Dixit,C Srimathi, “Smart Parking with Computer Vision and IoT Technology”, 2020 43rd International Conference on Telecommunications and Signal Processing (TSP),2020, https://doi.org/10.1109/TSP49548.2020.9163467.

[25] Jihoon Yang; Jorge Portilla“Smart parking service based on Wireless Sensor Networks” IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, 2012, https://doi.org/10.1109/IECON.2012.6389096.

[26] Stefano Bilotta“Predicting Free Parking Slots via Deep Learning in Short-Mid Terms Explaining Temporal Impact of Features” vol:12, Pgno: 101678 – 101693,2023, https://doi.org/10.1109/ACCESS.2023.3314660

[27] Wael Alsafery,Badraddin Alturki,“ Smart Car Parking System Solution for the Internet of Things in Smart Cities”, 2018 1st International Conference on Computer Applications & Information Security (ICCAIS),2018, https://doi.org/10.1109/CAIS.2018.8442004