

**REGION-BASED CONVOLUTIONAL NEURAL NETWORKS AUTOMATIC NUMBER PLATE RECOGNITION PARKING SYSTEM**

**IHAVI BRENDA IMBIAKHA**

**SCT221-C004-0288/2021**

**CLASS OF 2025**

**SUPERVISOR**

**DR. JUDY GATERI**

**Submitted in Partial Fulfilment of the Requirements of the Bachelor of Science**

**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY at JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**

**School of**

**PURE AND APPLIED SCIENCE/INFORMATION TECHNOLOGY at**

**JKUAT NAIROBI CBD CAMPUS**

# **Declaration and Approval**

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the research documentation contains no material previously published or written by another person except where due reference is made in the research documentation itself.

Student Name: **IHAVI BRENDA IMBIAKHA**

Admission Number: **SCT221-C004-0288/2021**

Student Signature: Date:

The Documentation of **Ihavi Brenda** has been reviewed and approved by **DR.** **Judy Gateri**

Supervisor Signature: Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **ACKNOWLEDGEMENT**

I take this opportunity to express my sincere gratitude to Jomo Kenyatta University and Technology for providing me with a platform to pursue my studies and carry out my final year project. I express my deep sense of gratitude to DR. Judy Gateri who is my supervisor for her continued help and guidance during the development of this proposal. I consider it a privilege and honour to express my sincere gratitude to all lecturers who were there to guide me throughout the tenure of this review. Finally, I would like to thank my partner, family and friends for the moral support they have given me during the completion of this work.

# **ABSTRACT**

The Region-based Convolutional Neural Network (RCNN)-Based Automatic Number Plate Recognition (ANPR) Parking System designed to streamline parking operations, enhance security, and improve overall efficiency at the Uchumi House building. The systems RCNN algorithm is set to detect and extract number plates from captured images which help in parking allocation and management hence improving the operations and workflow in the parking system. This system will help in improving efficiency and save on time in parking and control of traffic entering and leaving the building.

# **List of Abbreviations**

**ANPR**- Automatic Number Plate Recognition

**RCNN**- Region-based Convolutional Neural Networks

**AI**- Artificial Intelligence

**OCR**- Optical Character Recognition

**YOLO**- You Only Look Once

**DL**- Deep Learning

**IOT**- Internet of Things

**ML**-Machine Learning

**VMS-** Video Management System

**CTC**- Connectionist Temporal Classification

**DPS**- Digital Pressure Sensor

**RFID**- Radio Frequency Identification

**ASCII**- American Standard Code for Information Interchange

**LED**- Light Emitting Diode

**GPS**- Global Positioning System

**DFD**- Data Flow Diagram

**SVM**- Support Vector Machine

**API-** Application Programming Interface

Table of Contents

[**Declaration and Approval** ii](#_Toc199250210)

[**ACKNOWLEDGEMENT** iii](#_Toc199250211)

[**ABSTRACT** iv](#_Toc199250212)

[**List of Abbreviations** v](#_Toc199250213)

[**CHAPTER 1: INTRODUCTION** 1](#_Toc199250214)

[**1.1 Introduction** 1](#_Toc199250215)

[**1.2 Problem Statement** 2](#_Toc199250216)

[**1.3 Research Questions** 2](#_Toc199250217)

[**1.4 Objectives** 3](#_Toc199250218)

[**1.4.1 General objective** 3](#_Toc199250219)

[**1.4.2 Specific objective** 3](#_Toc199250220)

[**1.5 Justification** 3](#_Toc199250221)

[**1.6 Scope** 3](#_Toc199250222)

[**1.7 Limitations** 4](#_Toc199250223)

[**CHAPTER 2: LITERATURE REVIEW** 5](#_Toc199250224)

[**2.1 Introduction** 5](#_Toc199250225)

[**2.2 Review** 5](#_Toc199250226)

[**2.2.1 Review based on Automatic Number Plate Recognition** 5](#_Toc199250227)

[**2.2.2 Traditional Parking System** 6](#_Toc199250228)

[**2.2.3 Machine-vision based smart parking management systems** 7](#_Toc199250229)

[**Benefits** 9](#_Toc199250230)

[**Challenges** 9](#_Toc199250231)

[**Application** 9](#_Toc199250232)

[**2.2.4 Internet of Things based parking management systems** 10](#_Toc199250233)

[**Benefits** 13](#_Toc199250234)

[**Challenges** 13](#_Toc199250235)

[**Application** 14](#_Toc199250236)

[**2.2.5 Artificial Intelligence and Machine Learning based parking management systems** 15](#_Toc199250237)

[**Benefits** 17](#_Toc199250238)

[**Challenges** 17](#_Toc199250239)

[**Application** 18](#_Toc199250240)

[**2.2.6 Review on Parking Detection Techniques** 18](#_Toc199250241)

[**Sensor-based Techniques** 19](#_Toc199250242)

[**Position Based Techniques** 20](#_Toc199250243)

[**Vision-Based Techniques** 21](#_Toc199250244)

[**2.3 The Region-based Convolutional Neural Networks Model** 26](#_Toc199250245)

[**2.3.1 Convolutional Neural Networks** 26](#_Toc199250246)

[**2.3.2 Region-based Convolutional Neural Network** 26](#_Toc199250247)

[**Benefits of using R-CNN** 28](#_Toc199250248)

[**Challenges of using R-CNN** 29](#_Toc199250249)

[**Application** 29](#_Toc199250250)

[**CHAPTER 3: SYSTEM ANALYSIS AND DESIGN** 31](#_Toc199250251)

[**3.1 Introduction** 31](#_Toc199250252)

[**3.2** **Feasibility Studies** 31](#_Toc199250253)

[**3.2.1** **Technical Feasibility** 31](#_Toc199250254)

[**3.2.2** **Economic Feasibility** 31](#_Toc199250255)

[**3.2.3** **Operational Feasibility** 31](#_Toc199250256)

[**3.2.4** **Legal Feasibility** 32](#_Toc199250257)

[**3.3** **System Analysis** 32](#_Toc199250258)

[**3.3.1** **Data Collection** 32](#_Toc199250259)

[**3.1.2 System Requirements** 32](#_Toc199250260)

[**3.1.2.1 Functional Requirements** 32](#_Toc199250261)

[**Vehicle sensing** 32](#_Toc199250262)

[**Vehicle detection** 32](#_Toc199250263)

[**Number plate Localization** 33](#_Toc199250264)

[**Text extraction** 33](#_Toc199250265)

[**Data Logging** 33](#_Toc199250266)

[**3.1.2.2 Non-Functional Requirements** 33](#_Toc199250267)

[**Scalability** 33](#_Toc199250268)

[**Accuracy** 33](#_Toc199250269)

[**Adaptability** 34](#_Toc199250270)

[**Reliability** 34](#_Toc199250271)

[**Security** 34](#_Toc199250272)

[**3.4** **System Analysis Diagrams** 35](#_Toc199250273)

[**3.2.1 Use Case Diagrams** 35](#_Toc199250274)

[**3.2.2** **Data Flow Diagrams** 36](#_Toc199250275)

[**3.2.3 Activity Diagram** 37](#_Toc199250276)

[**3.2.4 Sequence Diagram** 38](#_Toc199250277)

[**3.3** **System Design** 40](#_Toc199250278)

[**3.3.1 Introduction** 40](#_Toc199250279)

[**3.3.2 System Architecture** 40](#_Toc199250280)

[**3.3.3 The Database** 41](#_Toc199250281)

[**Database Structure** 41](#_Toc199250282)

[**Relationships** 43](#_Toc199250283)

[**3.3.4 User Interface Design** 44](#_Toc199250284)

[**User Interface Layout components** 45](#_Toc199250285)

[**The User Interface Design Principles** 46](#_Toc199250286)

[**3.3.5 System Security** 46](#_Toc199250287)

[**Security Objectives** 47](#_Toc199250288)

[**Threat Assessment** 48](#_Toc199250289)

[**Security Measures** 48](#_Toc199250290)

[**3.4** **Conclusion** 49](#_Toc199250291)

[**CHAPTER 4: SYSTEM IMPLEMENTATION AND TESTING** 50](#_Toc199250292)

[**4.1 Introduction** 50](#_Toc199250293)

[**4.2 System Development Environment** 51](#_Toc199250294)

[**4.2.1 Hardware Requirements** 51](#_Toc199250295)

[**4.2.2 Software Requirements** 54](#_Toc199250296)

[**4.3 Dataset Preparation** 56](#_Toc199250297)

[**4.3.1 Data Collection** 56](#_Toc199250298)

[**4.3.2 Data Pre-processing** 57](#_Toc199250299)

[**4.4 Model Training and Fine-tuning** 61](#_Toc199250300)

[**4.4.1 Faster R-CNN Model Selection** 61](#_Toc199250301)

[**4.4.2 Training Process** 62](#_Toc199250302)

[**4.4.3 Model Evaluation** 63](#_Toc199250303)

[**4.5 Front-End Development and System Integration** 64](#_Toc199250304)

[**4.5.1 Overview** 64](#_Toc199250305)

[**4.5.2 User Interface Design** 64](#_Toc199250306)

[**4.5.3 API Integration and Backend Communication** 66](#_Toc199250307)

[**References** 70](#_Toc199250308)

[**APPENDIX 1: BUDGET** 74](#_Toc199250309)

[**Budget for Academic Purpose** 74](#_Toc199250310)

[**Client Budget Breakdown** 75](#_Toc199250311)

[**Assumptions and Justifications** 76](#_Toc199250312)

[**Conclusion** 77](#_Toc199250313)

[**APPENDIX 2: SCHEDULE OF PROJECT** 78](#_Toc199250314)

[**Project Phases** 78](#_Toc199250315)

[**Estimated time for phase development** 78](#_Toc199250316)

[**Tabulated Project Schedule** 79](#_Toc199250317)

[**Gantt Chart presentation for the schedule** 81](#_Toc199250318)

[**Conclusion** 82](#_Toc199250319)

# **CHAPTER 1: INTRODUCTION**

## **1.1 Introduction**

In today’s era, we live in a world that is fast-paced where time is a precious resource that we cannot afford to waste. In this regard, parking management has become a critical concern for businesses across the city. In light with this challenge, it is important to adopt modern smart parking solutions to fasten operations, meet customer’s needs hence improving their satisfaction, maximize on space utilization and saving on time.

This project proposal is crafted specifically for Uchumi house, a well-known building in Nairobi City located along Agakhan walk. The system aim is to simplify the parking experience for users, reduce congestion at the entrance, reduce the security’s operational burden and utilize available spaces efficiently.

I appreciate the fact that various automated parking systems have been developed to help in streamlining operations at parking lots. For instance, Fahad and Yazeed (2022) saw a need for easy, secure and efficient parking operations hence came up with a way to guide drivers to parking their cars quickly with products that can be too easy to be combined and integrated for a simple and efficient solution. Streamlining parking, both at Entrance-Exit and through the car park, making the parking much more efficient, speeding up the process and potentially freeing up spaces more quickly for further occupation. ALPR systems have been extensively studied and developed since the 1970s, primarily for traffic law enforcement and vehicle monitoring. According to Farag et al. (2004), ALPR systems generally consist of three stages: license plate detection, character segmentation, and optical character recognition (OCR) for extracting the text from the image. The accuracy and efficiency of these stages determine the overall effectiveness of the system.

Despite these advancements in technology, there exists a gap in user friendly applications and real time analytics leading to inefficiency in management. This gap hence provides an opportunity for a more comprehensive management system.

This system will be able to use sensors to sense motion of incoming vehicle and trigger the camera to capture an image of the front part of the vehicle where the number plate is located. The system will then detect the number plate in the image by use of faster RCNN (region-based Convolutional Neural Network). The text in the number plate will then be extracted by use of OCR (Optical Character Recognition) technology and the recognized number plate text is then checked for existence in a pre-registered vehicle database. Once the system verifies the number plate in the database, it proceeds to assign a specific parking spot according to the organization or company specified. The system then gives feedback to the user through a digital display on the exact parking space that was assigned to them. When leaving the building, the number plate will be scanned at the exit and the parking space will be freed from the system to accommodate other incoming users.

## **1.2 Problem Statement**

With the increasing number of employees and customer that flow in the building for work and services, parking management has become a significant challenge at the Uchumi building. At the moment, a manual parking system is still being used where users have to find parking lots manually leading to user dissatisfaction and time wastage. The current way of operation can’t handle the daily increase in traffic at the entrance efficiently leading to poor utilization of parking spaces, confusion in finding available parking spaces and long waiting time. Due to the absence of real-time information of available parking spaces, users find it difficult to quickly locate available spaces leading to frustration and confusion at the parking lot. This problem has led to overcrowding at the building’s entrance making traffic to stretch out to public roads which is a disruption of business activities around the area. It has also led to mismanagement of available packing spaces and has led to bad customer experience. If it is not dealt with urgently, it will cause a negative influence to the business due to reduced customer retention which will in turn affect profitability and reputation of the business. Given the criticality of the problem at Uchumi House, implementing an AI-powered system that provides real-time space availability and allocation will significantly improve the parking operations and enhance customer experience.

## **1.3 Research Questions**

1. What are the most accurate and efficient methods for implementing license plate recognition in parking systems?
2. How can an AI-based system allocate parking spaces efficiently based on real-time availability data?
3. How will the search time for parking be reduced?

## **1.4 Objectives**

### **1.4.1 General objective**

The general objective is for the smart parking system to be able to optimize the parking operation in the building by keeping track of available parking spaces at the parking lot which will ensure a seamless parking experience to users.

### **1.4.2 Specific objective**

1. Developing an automated license plate recognition system (Convolutional Neural Networks) that will recognize vehicles upon entry and record their number plate and allocate them with a parking space.
2. To apply algorithms that dynamically allocates parking spaces based on availability when vehicles enter the building.
3. To reduce the average time spent by drivers searching for parking space trough real time identification and allocation of parking lot.

## **1.5 Justification**

This research project is significant to the Uchumi house building as it will enhance efficiency and space utilization by real time allocation of parking lots to users hence reducing congestion and time wastage. Users will benefit from a streamlined and hassle-free experience when the parking process is automated due to reduced traffic congestion at the entrance. Automating the parking process will also make work easy for the security guards in finding parking lots for incoming employees and guests.

## **1.6 Scope**

This project involves the development and deployment of a License Plate Recognition (LPR) parking system utilizing Region-based Convolutional Neural Network (RCNN) technology to automate vehicle identification, entry and parking allocation. It will use RCNN-based algorithms to detect and recognize license plates in real-time, eliminating the need for traditional camera systems. This project will focus on improving parking efficiency at Uchumi House and reducing the need for manual intervention.

## **1.7 Limitations**

This project will be constrained by the cost that will be incurred in purchasing the hardware required like sensors. When using image recognition, a high level of accuracy is required in order to correctly capture the image of the license plates in different environmental conditions. Since the project is tailored for a building that doesn’t require parking payment, the payment feature which is a part of many parking systems will be excluded from the design of this particular system.

# **CHAPTER 2: LITERATURE REVIEW**

## **2.1 Introduction**

This chapter delves into available parking system solutions in the market today. The number plate recognition system is a more efficient way of providing fast and efficient parking solutions. My proposed system will be able to use sensors to sense motion of incoming vehicle and trigger the camera to capture an image of the front part of the vehicle where the number plate is located. The system will then detect the number plate in the image by use of faster region-based Convolutional Neural Network. The text in the number plate will then be extracted by use of Optical Character Recognition technology and the recognized number plate text is then checked for existence in a pre-registered vehicle database. Once the system verifies the number plate in the database, it proceeds to assign a specific parking spot according to the organization or company specified. The system then gives feedback to the user through a digital display on the exact parking space that was assigned to them. When leaving the building, the number plate will be scanned at the exit and the parking space will be freed from the system to accommodate other incoming users.

## **2.2 Review**

## **2.2.1 Review based on Automatic Number Plate Recognition**

There have been various studies being conducted recently regarding the Automated Number Plate system. The traditional way of parking where users or security guards could manually look for parking without the knowledge of availability has been out dated with the emergence of the new parking technologies.

ANPR is not limited to parking, it can be used in different fields that require the data of number plates to aid in performing their tasks. According to V. Gnanaprakash et al. (2020), Tracking individual vehicles became challenging as the automotive industry continued to increase dramatically daily. The authors proposed an automated vehicle tracking model that used roadside security cameras to track rapidly moving automobiles. They employed an effective deep learning (DL) model for object detection to address this subject.

Ravi Kumar et al. (2020) described significant groups of intelligent transportation and detection mechanisms (STDM) that composed of tools in which the system detects and mechanically reads the license identification of the vehicle Number Plate from digitally considered images. Automatic recognition of the license Number Plate was the development of distinguishing and transforming the pixel data of a digital image into plain texted data or ASCII code-based number plate. Using measured morphological operations, the authors implemented a method for detecting vehicle number plates from the photo. The primary motive of the developed model was to deploy and compose various morphological developments in which the license NP of a particular vehicle could be distinguished and translated effectively. The implemented model could detect the license Number Plate perfectly and rapidly from the vehicle’s picture.

On the other hand, Hengliang Shi et al. (2023) described the DL model for license NP position and detection in natural scenarios, which was recommended in a determination to address the subject of limited accuracy and speed through traditional methods. Firstly, they improved the channel care method of the you-only glanced once (YOLOv5) down sampling procedure. To reduce information loss due to sampling, site data was added to the ones, which might have increased the Model's robustness for feature extraction. The detector for finding license plates was made more effective and precise by reducing the number of metrics on the input side and setting only one class in the yolo layer. Gated recurrent units (GRU) and connectionist temporal classification (CTC) were used to complete the acknowledgment network.

As several vehicles increased, the number of traffic offences on the roads increased as well. Depending on the Indian traffic road system, keeping track of a violation from such an incident was laborious and difficult. However, M. Maheswari et al. (2020) came up with measures to be taken to monitor traffic infractions on the road by using a system to identify vehicles with license plates involved in speeding or breaking the law. In the context of vehicle license plate identification for innovative transportation structures, license plate position was a fundamental subject. This system was meant to improve security by detecting vehicles involved in crime activities.

## **2.2.2 Traditional Parking System**

Euro Parking Services Ltd (2024) defines traditional parking system as a process that involves motorists parking their vehicles using manual processes, such as waiting in line to collect parking tickets and paying upon departure. These systems offer basic facilities with limited features as compared to the modern ANPR systems.

**Benefits**

1. Traditional parking systems operate without advanced machinery, making them less expensive to implement.
2. Traditional parking systems rely more on manual operations rather than number plate recognition and digital mobile payments.
3. The parking process involves simple steps, making it easy for visitors or motorists to become familiar with and quickly learn the process.

**Challenges**

1. Motorists often have to wait in long queues to purchase a parking ticket, leading to lower productivity and decreased efficiency.
2. Manual management of parking systems increases the risk of errors in recording vehicle details and the likelihood of theft and vandalism due to limited security measures.
3. Lack of facilities and time-consuming tasks in traditional parking spaces can cause significant inconvenience for motorists, negatively impacting the customer experience.

### **2.2.3 Machine-vision based smart parking management systems**

According to (Hemalatha V, et al, 2023), Vehicle parking using number plate recognition using OCR (Optical Character Recognition) algorithm aims to automate the process of vehicle parking management in a parking lot or garage. The project involves the use of a camera to capture images of the license plates of vehicles entering and exiting the parking area, and an OCR algorithm to recognize the license plate numbers. The system is designed to compare the recognized license plate number with a database of registered vehicles to determine if the vehicle is authorized to park in the parking lot or garage. If the vehicle is authorized, the system will reserve the parking space or process payment automatically. The OCR algorithm is trained to recognize the characters on the license plate, and can handle variations in font, size, and color. The algorithm extracts the characters from the license plate image and converts them into digital text that can be processed by the system. This system can be further enhanced by integrating it with other parking management systems such as access control, reservation systems, and payment systems. It can also be used to monitor the occupancy of parking spaces in real-time, optimizing the parking lot layout, and maximizing the utilization of available space. The benefits of vehicle parking using number plate recognition using OCR algorithm includes reducing the time spent searching for a parking space, improving traffic flow, enhancing security and surveillance, and increasing revenue by minimizing parking violations and improving efficiency. This approach provides a smart and efficient way to manage parking lots and garages while providing a seamless customer experience. It can be implemented in various parking areas, including public parking lots, commercial parking lots, and private parking areas.

(Isarsoft GmbH, 2024) Argues that Automatic Number-Plate Recognition (ANPR), also referred to as Automatic License-Plate Recognition (ALPR), is a technology that uses optical character recognition (OCR) and image or video analytics techniques to read and capture the license plate information from vehicles. ANPR systems are commonly used in various applications, including law enforcement, toll collection, parking management, and traffic monitoring. According to Isarsoft, ANPR systems typically consist of security cameras or special ANPR cameras that capture images of vehicles' license plates, software that processes and analyzes the images, and a database that stores and matches the license plate information against known records. The technology can automatically extract the alphanumeric characters from the license plate image and convert them into text, enabling the identification of vehicles and the retrieval of relevant information such as vehicle registration details. The ANPR software can run directly on the camera, inside the video management system (VMS) or in a dedicated video analytics application.

Car Park perimeter security dramatically improves with effective automatic number plate recognition. When ANPR operates as the heart, or nucleus, of an overall car park security strategy, car park and property stakeholders reap multiple benefits from short-term executional efficiencies to longer-term strategic improvements. Not only can ANPR track entrance and exit times of all vehicles and site occupancy, but it can also deliver customized alerts to deliver meaningful real-time data to those who need it most. (Silicon Valley and Budapest, 2023) also points out that Modern-day car parks act as a natural barrier that offers perimeter security to see who’s on the premises before they enter any building. When strengthening this natural barrier with a powerful ANPR, management can boost security and efficiency without being perceived as overly fortified. Aside from identifying visitors before entering buildings, ANPR can help car park perimeter security also monitor how long each vehicle has remained on the property and ensure that scarce parking spaces are used appropriately.

(Ravi Kiran, et al, 2020) presented an image processing technique for Indian license plate detection and recognition with various conditions such as noisy environment, low light, non-standard license plate and cross angled situation. For the pre-processing they used several techniques such as gaussian smoothing, morphological transform and gaussian thresholding. For the segmentation they used contours and K-nearest neighbor algorithm for character recognition.

### **Benefits**

* Computer vision algorithms provide real-time, highly accurate detection of available parking slots.
* Machine vision-based systems use cameras and software to monitor multiple spaces at once hence reducing the installation and maintenance costs associated with physical sensors, making the system more scalable.
* These systems enhance the security of cities as the cameras used to detect parking spaces can detect criminal activities around the areas they are installed.
* Helps in streamlining access control, billing and enforcement eliminating manual inspections and paper work.
* These systems can be easily scaled to cover larger areas and cameras can also be upgraded to add new functionality.

### **Challenges**

* Initial cost is high as it requires installation of high-quality cameras and advanced software.
* Machine vision systems may have a challenge in adverse weather conditions leading to obstruction of cameras hence leading to reduced visibility which may affect the performance of the system.
* Use of these systems raise privacy concerns as it involves the collection of sensitive data like the number plates of vehicles.

### **Application**

According to (Marr, 2021) machine vision has the below existing examples in practice;

* Machine vision is used in hospitals for medical imaging and analyzing x-rays, CT scans and MRIs.
* Machine vision is used in industrial automation and manufacturing sector to automate quality control by inspecting products for defects, manage production lines by identifying objects, reading barcodes and tracking items in real time.
* Machine vision is used in retail for automating checkout solutions. These systems can recognize products, calculate totals and also detect suspicious behaviors within the store.
* Machine vision systems in agriculture help monitor crops, detect plant diseases, and manage harvests efficiently. These systems can analyze drone footage or images captured by cameras to assess the health of crops and optimize resource usage.
* Machine vision is widely used in security and surveillance to automatically monitor environments. These systems can detect abnormal behavior like theft, recognize faces, and track movements in real-time, improving safety in public spaces.
* Machine vision plays a crucial role in autonomous driving systems in that Cameras and sensors equipped with computer vision algorithms enable vehicles to perceive and understand their surroundings, identifying pedestrians, other vehicles, road signs, and lane markings hence aid to the safety of these vehicles.

## **2.2.4 Internet of Things based parking management systems**

According to (Eoin O’Connell et al, 2021), The Internet of Things (IoT) has come of age, and complex solutions can now be implemented seamlessly within urban governance and management frameworks and processes. For cities, growing rates of car ownership are rendering parking availability a challenge and lowering the quality of life through increased carbon emissions. The development of smart parking solutions is thus necessary to reduce the time spent looking for parking and to reduce greenhouse gas emissions. Thus, by use of sensors which are embedded in parking spots, data is sent to a centralized system that informs drivers about available spaces in real time, reducing the time spent searching for parking. This system is highly beneficial for urban areas with limited parking availability​. (Naphade et al, 2020) also emphasized on the use of smart parking and explained how it could be used to optimize the use of resources like fuel and reduce traffic congestion. He said that the systems would allow for dynamic pricing based on demand, which further encourages more efficient use of parking spaces​.

Journal of Physics Conference Series, (2023) wrote that a person named Soni proposed a smart parking system using IoT cloud incorporated system for indoor and outdoor parking areas. The system architecture uses Bluetooth to communicate with the application and ultrasonic sensor to detect indoor parking spot occupancy. Outdoor parking uses the bit mounts with attractive sensor module that is able to get Universal Subscriber Identify Module through Bluetooth correspondence. Based on data from multiple cameras, Vítek and Melničuk created a parking monitoring system to identify whether a parking place is occupied. The system is implemented using tiny camera modules built on a Raspberry Pi Zero, a computationally efficient occupancy detection algorithm based on the histogram of oriented gradients (HOG) feature descriptor and support vector machine (SVM) classifier, as well as a convolutional neural network (CNN) of database called PK-Lot, PK-Slots, and FEL-Slot that contains over 10,000 images in various weather conditions. The system should have the ability to supply occupancy information with greater than 90% accuracy at a rate of 10 parking spaces per second under a variety of circumstances.

The recent studies on smart parking implementations in different cities worldwide provide a broader perspective on the advancements and effectiveness of smart parking systems in urban environments. (Vesna, et al, 2024) provides an overview of how different urban centers have embraced smart parking. First, Barcelona has integrated various smart city initiatives, including smart parking systems that utilize IoT-enabled sensors and mobile applications. These systems provide real-time parking availability, reducing traffic congestion and enhancing urban mobility. The city’s Urban Mobility Plan aims to have over 80% of journeys made via sustainable modes by 2024.The SFpark project uses dynamic pricing based on real-time demand data. This approach optimizes parking space usage, reduces cruising for parking, and lowers greenhouse gas emissions. The integration of autonomous parking solutions with IoT technologies has further enhanced the efficiency and effectiveness of the city’s parking management. On the other hand, London utilizes Automatic Number Plate Recognition (ANPR) technology to monitor and manage parking spaces. This technology provides real-time data on parking availability and aids in enforcing parking regulations. The city’s smart parking system also integrates mobile payment options, streamlining the parking process for users and improving overall efficiency. This innovative approach combines IoT and deep learning to provide accurate and reliable predictions for parking availability, showcasing the potential of advanced technologies in improving urban mobility solutions. Amsterdam’s smart parking systems employ a combination of sensors and cameras to monitor parking space usage. Real-time data is provided to users via mobile apps, allowing for automated payment and reservation of parking spaces. These systems have significantly improved the city’s ability to manage high traffic densities and optimize parking resources. Singapore has developed a unified platform that integrates various parking systems across the city. This platform collects data from multiple sources, including IoT sensors and cameras, to provide real-time parking information. The system supports electronic payments and dynamic pricing to manage parking demand effectively. Integration of Internet of Things (IOT), Machine Learning (ML), and Artificial Intelligence (AI) in smart parking systems, aiming to address critical urban issues like traffic congestion and inefficient parking space utilization.

Nowadays, modern cities face traffic congestion due to parking space constraints. Two main factors for reduced navigable space are increased vehicles on fixed space and skewed demand for on-street parking. Switching to multi-level parking system addresses land scarcity, but results in increased slot-hunt time. (Maya, et al, 2021) talks of internet of things (IoT) based smart multi-level parking solution being employed using Wireless Sensor Networks (WSN). In this system, empty parking slots in each level are tracked and allocated to users on request. The sensor node authenticates incoming vehicles and can track possible speed violation using radar sensors. The motorized lift, capable of detecting floor levels using combination of digital pressure sensor (DPS), drops vehicle to allotted slot. The parking bill is generated considering the true usage time tracked using ultrasonic sensors and accurate timers. This system utilizes ESP-NOW protocol for inter-node communication and results show its suitability in real-time system owing to no-latency communication.

(Bante, et al, 2024) conduct study that presents a state-of-the-art Smart Parking System (SPS) that makes use of the convergence of artificial intelligence (AI) and the Internet of Things (IoT) to maximize parking space use, reduce traffic jams, and improve user experience. The system seeks to maximize parking space use, reduce traffic jams, and improve user experience by implementing smart sensors that reliably report parking availability, the Internet of Things infrastructure makes it possible to monitor parking spaces in real-time. These sensors connect easily to a centralized control system and are positioned thoughtfully throughout parking lots. The collected data is then processed using AI algorithms to predict parking space availability, considering historical usage patterns, special events, and dynamic factors influencing parking demand. Using machine learning techniques, the AI component of the system analyzes and predicts user behavior to provide individualized parking recommendations based on past preferences and restrictions of the present moment. As a result, the parking system operates more efficiently overall, people spend less time looking for parking, and the urban ecosystem becomes more ecologically friendly and sustainable. In addition, the Smart Parking System comes with an easy-to-use mobile application that lets drivers view parking information in real-time, get customized recommendations, and even reserve spots ahead of time. The application also facilitates seamless payment transactions, creating a hassle-free parking experience for users. The implementation of this IoT and AI-based Smart Parking System holds significant promise in transforming urban mobility, addressing traffic congestion, and promoting a sustainable and intelligent urban environment. The research presented herein includes a detailed analysis of system architecture, performance evaluations, and a discussion on the potential societal impacts of such an innovative parking solution.

(Shobayo, et al, 2020) presented a vehicle number plate recognition system which is an IoT based system having the higher sensor for taking the image of a vehicle and the image processing methods to locate, segment and detect the vehicle number from the number plate. It uses OpenCV for implementation along with different IoT related hardware such as higher sensor Raspberry Pi and other components.

### **Benefits**

* The convergence of Internet of Things (IOT), Machine Learning (ML), and Artificial Intelligence (AI) in smart parking systems will significantly enhance parking management by providing real-time data analysis and predictions.
* Adopting advanced technologies such as electromagnetic sensors, ANPR cameras, and mobile applications, can enhance its parking infra-structure and improve the overall quality of urban life.
* Sensors placed in parking spaces allow for real-time monitoring of parking availability, these sensors transmit data to centralized systems that drivers can access via mobile apps or digital signage, reducing the time spent searching for parking.
* Due to reduced search time of parking spaces, traffic congestion is drastically reduced as drivers are shown exactly where they can find free parking slots.
* It enhances user experience since everything is automated and they can access the data from anywhere on their mobile phones which enable them to reserve and even pay for parking from wherever they are.
* Automated monitoring space occupancy and ticketing leads to reduced overhead costs.
* Leads to reduction in carbon emission produced by vehicles when drivers are searching for parking spaces.
* Leads to less fuel consumption due to reduced time for searching parking spaces hence helps the drivers to cut on fuel consumption costs.

### **Challenges**

* Implementing this technology incurs a lot of cost due to installation of IoT sensors, cameras, and communication networks.
* IOT generates vast amounts of data and protecting these data from attacks and unauthorized access raises a big concern.
* There should be regular maintenance of IOT devices to ensure continuous connectivity and functionality of the system.
* Due to lack of interoperability among all Internet of things systems, design of a perfect system with less cost becomes a difficult job.

### **Application**

Apart from smart parking solutions, Internet of Things has been implemented in several other areas around the world as stated by (CSL DualCom Ltd, 2024). Some of these areas are listed below;

* Internet of things is being used to develop smart cities, where various systems are connected and share data to improve the efficiency of city operations. With regard to this particular aspect, Sensors can be used to detect leaks in the water system, saving both water and money. IoT can also be used to monitor energy consumption in real-time, allowing for more efficient use of resources. Connected cameras and sensors can help law enforcement identify and track criminals. Lastly, Sensors can be used to monitor air quality, water quality, carbon dioxide levels, temperature, and noise levels.
* Smart homes are becoming increasingly popular, as they offer a number of benefits, such as temperature regulation hence saving energy and money, monitoring the activities in the home, deterring burglars and alerting the homeowners to any potential threats. Automated tasks, like turning on the lights or opening the garage door, can make everyday life more convenient. These tasks can typically be done through mobile phones or remote technology. LED bulbs can also be controlled remotely, saving energy and money among many others.
* IoT can be used to track products throughout the supply chain, from a product’s manufacture to its delivery to the customer. This data can be used to optimize production by understanding where products are in the supply chain hence meeting demand. Customer service can be improved by tracking the location of products in the supply chain. Businesses can provide better customer service by offering accurate delivery estimates. IoT can also be used to track assets, like containers and pallets, throughout the supply chain which helps to ensure that goods are delivered on time and that nothing is lost or stolen. RFID tags can be used to track inventory, making it easier to keep track of stock levels and avoid stock-outs.
* The use of IoT in the healthcare industry has skyrocketed following the pandemic, with some of its uses including doctors monitoring patients at risk at home through use sensors, allowing them to take preemptive action if necessary. IoT has also been used to develop contact tracing apps which utilize Bluetooth technology to track when two people with the app are in close proximity to one another. The Internet of Things helps doctors streamline patient medical records and access, making real-time data available where and when it’s needed. Pharmaceutical manufacturing is being optimized by IoT as well. Tracking production in real-time means that issues can be caught and resolved quickly, preventing costly delays.
* Wearable technology has come on leaps and bounds thanks to IoT, it is being used in wearables like smartwatches to read messages, fitness and activity tracking. smartwatches offer users a range of services on one condensed device. The functionality of these devices is undeniable, whether you’re tracking the location of your children or counting the number of steps you’ve taken throughout the day. Virtual reality (VR) and augmented reality (AR) gear has become a hot property for gamers and tech enthusiasts. However, these technologies are also being used in a number of other industries, including healthcare, education, and retail. IoT helps with the location monitoring of elderly patients in hospitals and care homes, providing more support for caregivers.
* IoT can be used to increase the efficiency of agricultural production by use of smart irrigation systems which are used to reduce water usage, saving both water and money. RFID tags can be used to track livestock, making it easier to manage herds. IoT can also be used to monitor farm equipment, like tractors and combine harvester and the data can be used to optimize maintenance schedules and prevent downtime. Farmers on the other hand can use sensors to monitor soil moisture levels, optimize irrigation, and reduce water usage, the use of GPS can help them in mapping data to plan where to plant crops, apply fertilizer, and spray pesticides. This helps reduce the number of resources used and improves yield.

## **2.2.5 Artificial Intelligence and Machine Learning based parking management systems**

(Hwan, et al, 2021) introduces a smart parking lot management system using multiple cameras and artificial intelligence technique. When a vehicle enters a parking lot, it recognizes the vehicle number using embedded camera, tracks which parking space the vehicle is parked in, and updates parking space information. In addition, using a surveillance camera images, it has been also implemented to detect collision accidents that may occur while the vehicle is moving in the parking lot. Vehicle number recognition system uses OCR technique and is implemented on a Raspberry system. By managing the vehicle number recognized at the entrance of the parking lot as an Object ID, it is possible to effectively track the vehicle as a moving object inside the parking lot and finally identify the parking location. In order for accident detection, YOLO with CNN deep learning process is used. More than 500 possible collision images are trained in advance. Experimental results show that the detection accuracy of parking and accident detection increases as the number of training images increases. The accident detection needed more training images because it has more diversity. By using the smart parking system implemented in this paper, it is possible to effectively manage the vehicle's parking location, free space information and possible accidents. Using a cloud system, implemented system can provide drivers an integrated parking lot information over large areas.

Parking space management has become a critical challenge in urban areas due to increasing vehicle numbers and limited parking infrastructure. (Dahiya, et al, 2024) presents a comprehensive study of machine learning (ML) models in IoT-enabled environments focusing on proposing an ML-based model for predicting available parking space. The study evaluates the performance of various models including K-nearest neighbors (KNNs), support vector machines (SVMs), random forest (RF), decision tree (DT), logistic regression (LR), and Naïve Bayes (NB) based on “precision, recall, accuracy, and F1-score performance metrics”. The results obtained by implementing ML models on the data with 65% and 85% threshold values are compared to draw meaningful conclusions regarding their performance in predicting parking space availability. Among the evaluated models, random forest (RF) demonstrates superior performance with high precision, recall, accuracy, and F1-score values. It showcases its effectiveness in accurately predicting parking space availability in the IoT-enabled environment. On the other hand, models such as K-nearest neighbors (KNNs), decision tree (DT), logistic regression (LR), and Naïve Bayes (NB) show relatively lower performance in complex parking scenarios. The authors concludes that the use of advanced predictive models, particularly random forest, significantly enhances the accuracy and reliability of IoT-enabled parking management systems and also reduces the waiting time of the vehicles, leading to more efficient resource utilization, reduced traffic congestion in real-time scenarios, and better user satisfaction in the IoT-enabled environment.

### **Benefits**

* Machine learning can automate the most repetitive and mundane tasks, leaving the human workforce free to spend time on productive tasks that are potentially much more sophisticated and creative.
* By creating an AI robot that can perform complex tasks on our behalf, we can overcome many dangerous restrictions humans face. It can be utilized effectively in any natural or man-made calamity, whether going to Mars, defusing a bomb, exploring the deepest regions of the oceans, or mining for coal and oil.
* Machine learning algorithms derive insights out of massive data that would normally be difficult, if not impossible, for a human analyst to decipher. Finance is an area where, for instance, ML models can be used to predict stock market tendencies that aid investors in making informed decisions.
* One of the key benefits of Artificial Intelligence is round the clock availability. Many studies show humans are productive for only about 3 to 4 hours daily need breaks and time off to balance their work and personal lives. But AI can work endlessly without breaks and they think much faster than humans and perform multiple tasks simultaneously with accurate results.
* ML systems process and analyze data in real time, giving instant feedback to decision-making processes. For example, in applications like fraud detection by banks, timely intervention is possible without delay because the event is happening in real-time.
* Emotions inherently drive humans, while AI operates without emotional influence, maintaining an efficient and rational approach. One significant advantage of Artificial Intelligence is its lack of biased views, leading to more accurate and objective decision-making.

### **Challenges**

* To learn better, a machine learning model requires sufficient, voluminous, and high-quality data. Poor data results in poor model performance.
* Training complex ML models demands a significant power and resource pool that can be quite expensive and out of reach for a few organizations.
* Artificial Intelligence (AI) often lacks the intrinsic creativity of humans, which stems from emotional depth, abstract thinking, and imaginative processes. While AI can mimic creativity by generating art, music, or writing based on existing patterns, it doesn't possess genuine originality or the ability to think outside the box.
* The rise of AI and automation technologies poses a substantial risk to employment, particularly in industries reliant on routine and repetitive tasks.

### **Application**

Artificial Intelligence and Machine Learning has numerous applications available in the market as explained by (Biswal, 2024)

* Artificial Intelligence enhances robots' capabilities, enabling them to perform complex tasks precisely and efficiently. In industries like manufacturing, AI-powered robots can work alongside humans, handling repetitive or dangerous tasks, thus increasing productivity and safety.
* Natural Language Processing (NLP) is an AI field focusing on interactions between computers and humans through natural language. NLP enables machines to understand, interpret, and generate human language, facilitating applications like translation, sentiment analysis, and voice-activated assistants.
* Face recognition technology uses AI to identify and verify individuals based on facial features. This technology is widely used in security systems, access control, and personal device authentication, providing a convenient and secure way to confirm identity.
* AI-powered chatbots provide instant customer support, answering queries and assisting with tasks around the clock. These chatbots can handle various interactions, from simple FAQs to complex customer service issues.
* Machine Learning transforms the entertainment industry by personalizing content recommendations, creating realistic visual effects, and enhancing audience engagement. It can analyze viewer preferences, generate content, and create interactive experiences.

## **2.2.6 Review on Parking Detection Techniques**

In addition to allocating parking vehicles to customers, the RCNN based ANPR system help its users to detect availability of parking spaces by displaying both the occupied and available parking slots. There are other techniques that exist which are used to detect the availability of parking spaces. Below are some of the techniques that are effective in such a scenario:

### **Sensor-based Techniques**

This technique provides real time information about the availability of parking spots. Below are some of the sensors used in parking slot detection;

1. **Ultrasonic Sensors**

Xion, et al, (2019) argues that these sensors are used to detect the presence of automobiles in parking spaces. ultrasonic sensors are usually installed on the ceiling or walls of the parking garage. The parking control system receives a signal from the ultrasonic sensor when a car pulls into a parking spot and updates its database to indicate that the parking space is occupied. Ultrasonic sensors are inexpensive and easy to install.

1. **Infrared Sensors**

Infrared sensors work by emitting infrared radiation and by detecting the reflection of this radiation from a vehicle. Kadir, et al, (2020) explains that when a vehicle enters the field of view of the sensor, the infrared radiation emitted by the sensor is reflected by the vehicle and detected by the receiver. The distance between the sensor and the vehicle can be calculated based on the time it takes for the infrared radiation to travel to the vehicle and back to the sensor. This data is used to assess if a parking place is occupied or not. Infrared sensors are particularly useful in outdoor parking lots, where lighting conditions can vary significantly throughout the day

1. **Magnetic sensors**

When a vehicle enters a parking space, it creates a disturbance in the magnetic field, which is detected by the detector. The magnetic sensor can determine whether a parking space is occupied or not based on the strength and duration of the disturbance in the magnetic field. A central control system that can offer real-time information on parking space occupancy can receive this information and deliver it to the appropriate parking spaces. This is according to Abidin, et al, (2020).

1. **Anisotropic Magneto Resistance (AMR) Sensors**

The AMR sensors are commonly used for detecting changes in magnetic fields. Floris, et al, (2020) indicates that the basic principle behind AMR sensors is that they have a varying electrical resistance based on the direction of the magnetic field passing through them which makes them ideal for detecting the presence of a nearby metallic object. AMR sensors can be placed at regular intervals in a parking slot and connected to a central control unit. When a vehicle pulls into a parking spot, the metal in the vehicle disturbs the magnetic field surrounding the sensor, changing its electrical resistance. This change is detected by the control unit, which then registers the presence of a vehicle in the corresponding parking space.

1. **Light Detection and Ranging (LiDAR)**

LiDAR-based parking detection systems also provide real-time data on parking availability, which can be used to guide drivers to available spots and help reduce the time spent circling the lot in search of a parking space. According to Hassan, et al, (2021), this technique uses laser beams to measure the distance between the sensor and the target object, creating a 3D map of the surrounding environment. LiDAR is used to monitor the movement of vehicles in and out of parking spots

1. **Inductive Loop Detector Sensors**

These sensors are placed in the ground and work by detecting changes in the magnetic field caused by the presence of a vehicle. Allbadi, et al, (2021) outlines that the sensors consist of a loop of wire embedded in the ground which is associated with a control system that recognizes modifications in the magnetic field brought on by the presence of a car. The loop senses the disruption in the magnetic field caused when a car pulls into a parking spot. The control unit can determine whether a parking space is occupied or not based on the strength and duration of the disturbance in the magnetic field. A central control system that can offer real-time information on parking space occupancy can receive this information and deliver it to the appropriate parking spaces.

### **Position Based Techniques**

Kumar, et al, (2023) listed several techniques which are based on the position in which sensors are placed, the techniques he outlined are as follows;

1. **Surface-Mounted Sensors**

This technique has its sensors installed on the surface of the parking space and are used to detect the presence of a vehicle parked above it. They are often used in indoor parking garages and can be easily installed without requiring any excavation or drilling.

1. **Buried Sensors**

The sensors are positioned underneath the parking space’s surface and look for vehicles. They are frequently utilized for outdoor parking.

1. **Wireless Sensor**

It uses wireless technology to transmit data about parking availability to a central system. These sensors can be surface-mounted or buried and can be easily installed without requiring any wiring or cabling.

1. **Multi-Level Sensors**

They are used in multi-level parking garages to monitor parking availability in real time and to monitor the presence of vehicles on each level. They can be surface-mounted or buried and are often connected to a central system that displays parking availability to drivers.

### **Vision-Based Techniques**

Meng, et al, (2020) explains this technique as a category that contains a combination of machine learning algorithms and computer vision techniques to detect parking spaces is used, which can be marked with lines or other indicators. This algorithm can analyze the images captured by the cameras and identify parking spaces and their availability. In order to extract patterns and characteristics from massive volumes of data, deep learning approaches employ artificial neural networks with several layers. Deep learning algorithms can be used to automatically learn and detect parking slots in photos or videos in the context of parking slot detection. He goes further and delves in the different approaches for vision-based parking detection.

* **Object Detection Based**

It is a popular category of vision-based parking detection techniques. Some commonly used object detection-based approaches for parking detection are given below:

**YOLO (You Only Look Once)** is an object detection algorithm that can detect objects in real time by dividing the image into a grid of cells and predicting the class and location of the object in each cell. This algorithm can detect multiple objects in an image and has been applied to parking detection tasks with high accuracy and speed.

Single Shot Detector (SSD) is a single-shot object identification system that uses several convolutional feature maps of various scales to predict the kind and location of objects. SSD can detect objects of different sizes and aspect ratios and has been applied to parking detection tasks with high accuracy and efficiency. According to Bochkovskiy, et al (2020), Yolo has undergone several iterations listed below in order to enhance its performance;

**YOLOv2/YOLO9000 (2017):** Introduced batch normalization and anchor boxes for improved speed and accuracy

**•YOLOv3 (2018):** Added multi-scale predictions and residual connections for better detection across various sizes.

**•YOLOv4 (2020):** Enhanced with the CSP Darknet back-bone and advanced training techniques, achieving higher precision.

**•YOLOv5 (2021):** Focused on usability, scalability, and deployment ﬂexibility with various model sizes.

**•YOLOv6 (2022):** Optimized for edge devices with im-proved backbone and attention mechanisms.

**•YOLOv7 (2023):** Employed Auto ML techniques for dynamic model optimization, enhancing adaptability

**•YOLOv8 (2023):** Incorporated a transformer-based back-bone for better detection in dense scenes.

**•YOLOv9 (2024):** Utilized adversarial training to improve robustness against variations.

**•YOLOv10 (2024):** Implemented real-time feedback loops for dynamic adjustments, boosting accuracy.

These enhancements have established YOLO as a versatile and powerful option for real-time object detection

**Faster R-CNN** is a state-of-the-art object detection algorithm that uses a region proposal network (RPN) to generate object proposals and a convolutional neural network (CNN) to classify the proposals and refine their bounding boxes. Faster R-CNN has shown promising results in detecting cars in parking lots, and it can handle different parking lot layouts, lighting conditions, and car orientations.

**RetinaNet** is a recent object detection algorithm that uses a novel focal loss function to address the class imbalance problem in object detection tasks. This algorithm can detect objects with high precision and recall and has been applied to parking detection tasks with promising results.

**Mask R-CNN** is a variant of Faster R-CNN that can also predict object masks in addition to bounding boxes and class labels. Mask R-CNN has been applied to parking detection tasks to segment and track cars in parking lots.

* **Background Subtraction Based**

This approach uses the difference between the current image and the background image to detect moving objects. Some frequently used background subtraction-based approaches for parking detection are as follows:

**Gaussian Mixture Model (GMM)** mentioned by Patel, et al (2020) as a widely used background subtraction algorithm that models the pixel intensities of the background as a mixture of Gaussian distributions. GMM can adapt to changes in the scene and can detect moving objects in real time. GMM has been applied to parking detection tasks to detect cars in parking lots and estimate parking space occupancy.

Padmasiri, et al, (2020) gives the **Adaptive background subtraction methods** as another technique in this category which updates the background model continuously based on the current image and the previous background model. This method can handle gradual changes in the scene and can detect moving objects in real time. Adaptive background subtraction methods have been applied to parking detection tasks to detect cars in parking lots and estimate parking space occupancy.

**ViBe** is a background subtraction algorithm that uses a pixel-based sampling strategy to detect moving objects in the scene. ViBe can learn the background model from a few frames and can detect objects with low computational cost. This algorithm has been applied to parking detection tasks with promising results.

* **Feature Based**

This approach works by extracting features from the image, such as edges, corners, and texture, and use these features to detect parking spaces and estimate occupancy. Agrawal, et al (2020) outlines some commonly used feature-based approaches for parking detection are as follows:

**Hough Transform** is a feature-based approach that detects straight lines in the image. In parking detection tasks, Hough Transform can be used to detect the edges of parking spaces and estimate their orientation and position. Hough Transform can also be combined with other techniques such as color segmentation and contour detection to improve its accuracy and robustness.

**Haar-like features** are rectangular features that can be used to detect edges and corners in the image. Haar-like features have been applied to parking detection tasks to detect parking spaces and estimate their occupancy and can also be combined with machine learning techniques such as AdaBoost and SVM to improve their accuracy and efficiency.

Another approach is the **Local Binary Patterns (LBP)** which is a texture-based feature extraction method that can detect local patterns in the image. LBP can be used to detect the texture of parking spaces and estimate their occupancy and can be combined with other techniques such as edge detection and clustering to improve its accuracy and robustness.

Lastly, the **Scale-Invariant Feature Transform (SIFT)** is a feature extraction technique that can find and match important image points. SIFT can be used to detect the corners and edges of parking spaces and estimate their occupancy. SIFT can be combined with other techniques such as contour detection and clustering to improve its accuracy and robustness.

* **Deep Learning Based**

Rahman, et al, (2020) analyses this approach for parking detection as one that involves training a neural network model to detect parking spaces and estimate their occupancy. These approaches typically involve using convolutional neural networks (CNN) to extract features from the input image and make predictions about the occupancy of parking spaces. The authors listed the following as examples in this category:

**Deep convolutional neural networks (DCNNs)** which are a type of neural network that can be used for parking detection tasks using multiple layers of convolutional filters to extract features from the input image and predict the occupancy of parking spaces. DCNNs can be trained on large datasets of parking lot images to improve their accuracy and robustness.

**Transfer learning** is a technique that involves a pre-trained neural network model for parking detection tasks and can be used to leverage the knowledge learned by the pre-trained model on large datasets of images and apply it to parking detection tasks. Transfer learning can be used with various neural network models including VGGNet, ResNet, and InceptionNet.

In addition to the above techniques, there are several ways presented by different authors that give more ways used and proposed for parking space availability in parking facilities. Choi and Do, (2020), proposed a context-based parking slot identification technique that was motivated by how a human driver locates a parking space. This technique uses two deep network modules, i.e., the parking context recognizer and the parking space detector. They also suggested new assessment criteria for parking slot identification, which reflects whether a car can fit in the discovered parking space. For which, they used Fish-Eye Image Dataset. Using MobileNetV2, VGG-16, and Resnet-50, the classification accuracy was 98%, while the orientation error ranged from 1.3 to 4.1 degrees.

Chen et al, (2020) suggested an extendable process called FakePS to help with the training of parking slot identification models using artificial data. One can construct numerous simulated parking settings with FakePS and then automatically gather labelled surround-view photos. This pipeline makes it possible to gather lots of artificial surround-view photos from unity scenarios. Using the unlabeled real photos as a starting point, the proposed parking slot constancy loss refines the synthetic images while maintaining the parking spaces.

Another approach that was suggested by Suhr and Jung, (2021) was the approach for images from around view monitors (AVM) a one-stage trainable end-to-end parking space recognition. The suggested approach makes use of a convolutional neural network (CNN) to simultaneously gather global information (entry, type, and occupancy of parking slot) and local information (location and orientation of junction) and integrates them to detect parking slots with their characteristics. Using a grid to divide up an AVM image, this method uses CNN to extract features. The global and local information of the parking space for each grid cell is produced by convolution filters when applied to the recovered feature map. By integrating local and global parking space data using non-maximum suppression (NMS), final detection results are produced.

Zhihua Chen et al, (2021) proposed a multi clue recovery model-based approach for reconstructing parking places called the Generative Parking Spot Detection. In the suggested method, the parking space is first geometrically disassembled to designate the locations of its corresponding corners. The next step is to locate corners from a ground image taken by a car’s camera using a micro target recognition network. In order to appropriately recover a trustworthy true parking location, the fully pairing map was corrected using the multi clue model after these steps. The experimental findings showed that the recommended method outperformed a number of other current algorithms in terms of accuracy, which may reach more than 80% in most test circumstances.

De-Hui Jian and Chang-Hong Lin, (2020) also presented their idea of a line semantic segmentation model and a point semantic segmentation model based on multi-task learning that were linked. In the post-processing stage, the coordinates of the real parking spaces were found using the proposed models, which also produced images of the entrance line and corner center points. The model achieved an f-measure of 96.06% and a precision of 99.40%.

These are among the reviews of parking detection among others which were identified to be in use in various parking facilities to detect parking slots.

## **2.3 The Region-based Convolutional Neural Networks Model**

### **2.3.1 Convolutional Neural Networks**

Zhang, et al (2022) defines Convolutional Neural Networks as a deep learning model designed for data processing and analysis in computer vision. In this model, CNN is effective in image recognition and object detection. The authors give the below characteristics of RCNN;

**Characteristics of Convolutional Neural Networks**

* **Convolutional Layer:** The core that builds block of a CNN is the convolutional layer which extract features from images by applying a set of filters or kernels to the input image. These filters slide across the image, performing dot products between their weights and the corresponding pixels in the input image.
* **Pooling Layer:** Pooling layers are known to be used to reduce the spatial dimensions of feature maps, making the network more computationally efficient and reducing the risk of overfitting. Common pooling operations include max pooling, average pooling, and L2- norm pooling.
* **Fully Connected Layer:** Fully connected layers are typically used in the final stages of a CNN to classify images or predict object locations. These layers take the flattened output of the convolutional and pooling layers and connect each neuron in the previous layer to each and every neuron in the current layer.

### **2.3.2 Region-based Convolutional Neural Network**

Region-based Convolutional Neural Network (R-CNN) is a type of deep learning architecture used for object detection in computer vision tasks. RCNN was one of the pioneering models that helped advance the object detection field by combining the power of convolutional neural networks and region-based approaches. According to Petru (2023), RCNN paved the way for subsequent innovations in object detection, including Fast R-CNN, Faster R-CNN, and Mask R-CNN, each building upon and enhancing the capabilities of its predecessor. To grasp the nuances of these advanced R-CNN variants, it is essential to establish a solid foundation in the original R-CNN architecture.

The number plate recognition parking management system which is my proposed system will use the R-CNN model. First, the sensors at the entrance of the building will sense motion of incoming vehicle and trigger the camera to capture an image of the front part of the vehicle where the number plate is located. The system will then detect the number plate in the image by use of faster R-CNN which will employ an algorithm like selective search to generate region proposals within the image which identify where the number plate might be located. Each isolate image is then processed through a pretrained Convolutional Neural Networks (CNN) to extract features that help in distinguishing the license plate from the rest of the vehicle, in his case the characters of the Number Plate. The detected license plate region is segmented into individual characters by use of the Optical Character Recognition (OCR) technology and the recognized number plate text is then checked for existence in a pre-registered vehicle database. Once the system verifies the number plate in the database, it proceeds to assign a specific parking spot according to the organization or company specified. The system then gives feedback to the user through a digital display on the exact parking space that was assigned to them. When leaving the building, the number plate will be scanned at the exit and the parking space will be freed from the system to accommodate other incoming users. Petru (2023) outlines the steps that are underwent in RCNN and how it works.

1. **Region Proposal**

RCNN divides the input image captured by the camera into multiple regions referred to as region proposals. This part is responsible for generating a set of potential regions in the image that is likely to contain the number plate. These region proposals are usually generated by methods like selective search.

1. **Feature Extraction**

Each region proposal is then resized to a fixed size and passed through a pre-trained CNN to extract features. This CNN acts as a feature extractor, producing a fixed-length high-dimensional feature vector for each region representing the content of the region proposal.

1. **Object Classification**

The extracted feature vectors from the region proposals are fed into a separate machine learning classifier for each object class of interest. R-CNN typically uses Support Vector Machines (SVMs) for classification. For each class, a unique SVM is trained to determine whether or not the region proposal contains an instance of that class.

1. **Bounding Box Regression**

In addition to classification, R-CNN also performs bounding box regression. For each class, a separate regression model is trained to refine the location and size of the bounding box around the detected object which helps improve the accuracy of object localization by adjusting the initially proposed bounding box to better fit the object's actual boundaries.

1. **Non-Maximum Suppression (NMS)**

After classifying and regressing bounding boxes for each region proposal, R-CNN applies non-maximum suppression to eliminate duplicate or highly overlapping bounding boxes. Non-Maximum Suppression ensures that only the most confident and non-overlapping bounding boxes are retained as final object detections.

### **Benefits of using R-CNN**

* R-CNN provides accurate object detection by leveraging region-based convolutional features.
* R-CNN models can handle objects with different sizes, orientations, and scales, making them suitable for real-world scenarios with diverse objects and complex backgrounds.
* It is a versatile framework that can be tailored to suite specific needs.
* R-CNN can be adapted to a wide range of object detection tasks by simply retraining the model on the desired dataset. For instance, R-CNN can be applied to medical imaging, autonomous driving, and retail surveillance after tuning the model for each domain.
* he R-CNN architecture is modular, meaning it can integrate different kinds of region proposal algorithms and classifiers. This flexibility allows for experimentation with different configurations to improve performance for specific tasks.

### **Challenges of using R-CNN**

* R-CNN is computationally intensive as it involves extracting region proposals, applying a CNN to each proposal, and then running the extracted features through a classifier. This multi-stage process can be slow and resource-demanding.
* Due to its sequential processing of region proposals, R-CNN is relatively slow during inference.
* R-CNN may generate multiple region proposals that overlap significantly, leading to redundant computation and potentially affecting detection performance.

### **Application**

* R-CNN models are applied in autonomous vehicles to detect objects like pedestrians, road signs, and other vehicles. Object detection is critical for self-driving cars to navigate safely and make informed decisions in real-time. An example is the Tesla’s Autopilot driving platforms use variants of R-CNN to detect lanes, traffic signs, and obstacles, assisting in decision-making for navigation and safety.
* R-CNN models have been utilized in medical image analysis for detecting tumors, lesions, and other abnormalities in medical scans. The ability of R-CNN to localize objects with high accuracy makes it a valuable tool in radiology, where precise detection is essential for diagnosis. cancer detection is accomplished by localizing regions in mammograms or CT scans where tumors may be present, aiding doctors in early diagnosis and treatment planning.
* In security and surveillance systems, R-CNN is applied for facial recognition, crowd detection, and tracking objects or people. The model is used to detect and track moving objects or individuals across video frames, providing valuable data for monitoring activities and ensuring safety.
* R-CNN is applied in retail environments for inventory management, where it helps in recognizing and counting products on shelves. It is also used for monitoring customer behaviors, such as tracking foot traffic and identifying patterns in shopping.
* R-CNN models are also applied in robotic vision systems, where robots need to detect and interact with objects in their environment. This includes picking and placing tasks in manufacturing, as well as autonomous robots navigating through spaces while avoiding obstacles.

**2.4 Conclusion**

The literature has covered multiple aspects, from the evolution of traditional parking methods to modern automated systems that utilize machine vision, AI, and IoT. It has also explored the various advancements and research contributions to number plate recognition (NPR) systems within parking management where optical character recognition (OCR) and machine learning models emerged as the most prominent. These technologies offer a high degree of accuracy in identifying license plates for applications such as automated entry/exit and billing processes. The use of deep learning approaches, such as Convolutional Neural Networks (CNNs) and Relational Neural Networks (RNNs) have improved the accuracy and speed of detection, even under challenging conditions like varying lighting and weather​ (Shweta, et al, 2022).

The use of IoT-based solutions, which offer real-time data processing and analytics to optimize parking operations has also been emphasized. Through sensors, cameras, and mobile apps, IoT platforms provide seamless integration for vehicle tracking, space allocation, and reservation systems. These advancements reduce congestion and provide greater convenience to drivers​. However, there are om key challenges that have been highlighted in data privacy, high implementation costs, and the need for interoperability between various system components. On a positive side, the integration of AI and machine vision has enhanced the security and operational efficiency of NPR systems, allowing for better decision-making through data analytics. As cities move toward smart parking solutions, the role of automated license plate recognition (ALPR) systems will likely continue to expand, offering a scalable and more sustainable approach to parking management.

In conclusion, it has come to my attention that number plate recognition systems have become a critical component of modern parking solutions. However, further research is necessary to overcome technical limitations and ensure that these systems can be efficiently scaled while maintaining privacy and security compliance. For this reason, my proposed system will align in this and provide a desired solution customized for Uchumi House building.

# **CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

## **3.1 Introduction**

The traditional parking system at Uchumi House has many challenges that are visible and cause frustration to employees and even customers. These challenges include traffic at the entrance of the building, difficulty in parking slot availability detection, security guards being overwhelmed causing fatigue among others. My proposed system will use Region based Convolutional Neural Networks (RCNN) for number plate detection, OCR for text extraction which will help in parking allocation and management.

## **Feasibility Studies**

Feasibility studies of the RCNN based ANPR parking system is performed to evaluate whether it is viable and its possibility to perform as expected. This study assesses the technical, operational, economic and legal aspects of the proposed system.

### **Technical Feasibility**

The RCNN based ANPR system is technically feasible as it consists of new technological hardware and software resources including the high-resolution Cameras, GPUs, Power backup systems, and sufficient storage devices which will aid in the system implementation. The availability of open-source training datasets and tools for developing the RCNN model. The system also allows for scalability allows for upgrades and transition in upcoming and better architectures.

### **Economic Feasibility**

Economically, the RCNN based ANPR parking system is cost-effective as monitory resources are effectively allocated. Availability of open-source resources such as training models reduces the cost of having to purchase one. The fact that processes will be automated guarantees a reduction in labor which will lead to a cut in labor cost. The System will also offer a high Return on Investment as it will reduce time wastage and save customers on fuel used to find parking as well as time.

### **Operational Feasibility**

The RCNN based ANPR parking system is highly operational as it will improve the operations of the Uchumi House building by minimizing traffic and offering a smooth user experience of the system. Automating the workflow in parking slot allocation by number plate recognition increases efficiency and reduces manual operation. The user interface is simple and easily understandable to the security guards and other users who will be interacting with it hence will need minimal training.

### **Legal Feasibility**

The RCNN based ANPR parking system abides by the set standards and regulations as it follows the data protection laws that secures personal data by having secure storage and having clear terms of use and consent mechanisms to ensure adherence to the laws of the land.

## **System Analysis**

### **Data Collection**

The decision of coming up with ANPR parking system came about after visiting the Uchumi House building and monitored how the current parking system was operating. I observed the system operation including vehicle flow, parking allocation and even how the guards perform their duties. The inefficiency of the current system and the fatigue it causes to its users and even strain in operation was a motivating factor to the proposed system.

I went on to interrogate one of the guards on how they feel about the current system and gained some insights on the loopholes that are present in the system which gave me more ideas on how to design a system that could meet their specific needs.

### **3.1.2 System Requirements**

System requirements are important to ensure effective functioning of both the hardware and software application. The project system requirement includes the functional as well as non- functional aspects. These requirements are clearly articulated statements of what the system must be able to do in order to satisfy the customers’ needs.

### **3.1.2.1 Functional Requirements**

### **Vehicle sensing**

The system will sense the presence of incoming vehicles at the entrance by motion sensors that will be placed underground. This vehicle sensing mechanism will help to trigger the cameras to capture an image of the vehicle.

### **Vehicle detection**

The system captures an image of incoming and outgoing vehicles in real time using cameras installed at the entry and exit points. The image captured in this process will help in the detection of the region of interest where the number plate is located for more analysis.

### **Number plate Localization**

At this point, the system detects the region of interest where the number plate is located using the RCNN. RCNN produces many region proposals which could be potential locations where the number plate could be located. The system will then refine the detected number plate by surrounding it with bounding boxes.

### **Text extraction**

The system will then use Optical Character Recognition (OCR) to extract alpha numeric text from the localized number plate. A string of characters representing the number plate will be produced and later be used for parking allocation.

### **Data Logging**

The system shall save the details of each detected number plate in the database and allocate each vehicle with a parking slot according to the provided details. When the vehicle exits the building, the parkin slot will be released to accommodate another incoming vehicle. Data logging will also help in maintaining a record of all vehicles that enter the building on a daily basis.

### **3.1.2.2 Non-Functional Requirements**

### **Scalability**

The Number plate recognition system is designed in such a manner that it can support a sizable user base owing to future growth with its widespread use. The system needs to be able to handle increasing vehicles, locations and cameras with no performance degradation. Scalability considerations are well addressed by the architecture that is made to scale resources efficiently and cope easily with increased user requirements. This will ensure that the system remains steadfast and capable to adjust in order to meet the changing needs of the owners of the building who are the customers. It also ensures that the system accommodates business growth and increased load efficiently.

### **Accuracy**

Accuracy refers to the ability of the system to yield correct and precise results. To ensure accuracy, my proposed system is designed in a way that it should be able to detect a vehicle and localize its image properly for preprocessing. The system also ensures that it captures all the vehicles information in a frame without missing on anything and not detecting non vehicle objects.

During localization, the system is set to determine the exact position of the number plate on the detected vehicle by ensuring that the bounding box around the number plate is precise for accurate character extraction that happens in the next step.

The system ensures accurate extraction of alphanumeric characters from the localized number plate region by use of OCR and ensures there are no or minimal errors.

### **Adaptability**

Adaptability refers to the ability of the system to function effectively and efficiently in varying scenarios like geographic locations, weather conditions and number plate variations. My proposed system will handle different variations of number plate formats to ensure that it accommodates different guests accessing the building. It will also perform well under different environmental conditions to ensure accurate detection of the number plate.

The system is also designed to ensure that it adapts to existing and future hardware and software and also ensure high performance even in the event of increase in workload.

### **Reliability**

This aspect refers to the ability of a system to perform its intended functions efficiently over a period of time without unexpected failure. My proposed system will ensure a 99.9% uptime to ensure continuous operation. The system ensures consistency under the same conditions in that it produces same results when scanning the same vehicle at multiple scans.

The system is designed to detect and recover from failures with minimum impact on performance ensuring that there is minimal downtime and high data integrity.

### **Security**

Security is a vital aspect of my proposed system to ensure that data, the system and its users are protected from all kinds of threats. Securing this system brings about integrity, confidentiality and availability of the system. To ensure that security measures are employed, only authorized users are permitted to use the system and access data by use of role-based access.

Users of the systems should log in using their usernames and passwords in order to gain access to the system.

## **System Analysis Diagrams**

System analysis diagrams are graphical representations used analyse different aspects of a system, such as its processes, data flows, and relationships between components, which help in understanding the system's structure, functionality, and how it interacts with its environment.

### **3.2.1 Use Case Diagrams**

Use Cases are visual representations showcasing user interactions with the system, defining functionalities and interactions between actors and the system. They portray the different types of users of a system and the case. The use case diagram below shows a use case diagram describing the actors and the activities involved in the operation of the Automatic Number Plate Recognition System (ANPR)

Figure 1: Use Case Diagram

### **Data Flow Diagrams**

A data flow diagram is a graphical representation of the flow of data through processes and entities. In this proposal, the data show diagram will show how data will be captured and fed to the system, the flows it will follow to the processes it will undergo and eventually how it will be stored. The diagram below shows the DFD for the automated number plate parking system.

Security Guard

ANPR System

Output

Vehicle information Parking Availability log

Space allocation

Vehicle Database

Figure 2: Data Flow Diagram

### **3.2.3 Activity Diagram**

An activity diagram is a Unified Modelling Language (UML) diagram that models the workflow or flow of control through a system. It is a visual representation of the steps or activities involved in a process or use case, showing the sequence of actions, decisions, and transitions between them. In the RCNN based ANPR system, the system begins when an image of the vehicle is captured and fed into the system. The system then analyzes the image and pre-processing is done to localize the number plate and afterwards the system detects the number plate. The OCR is then applied to extract the characters of the numberplate which aid to parking lot allocation by displaying the parking position. This workflow shows how the activities flow in the system.



Upload image

Pre-processing

Allocate Parking

Display Text

Extract Number Plate

Detect Number Plate

No

Yes



Figure 3: Activity Diagram

### **3.2.4 Sequence Diagram**

The sequence diagram is a visual representation that is used to represent the dynamic behavior of the Number Plate Recognition System and provides a clear step by step representation of how different components in the system interact over a period of time to ensure accurate detection and recognition of vehicle number plates. The RCNN based ANPR is designed to ensure a continuous stream of image capture from cameras, processing and producing relevant outputs. The sequence diagram shows the flow of data and the smooth transitions between the components that are involved.

ANPR Parking System Sequence Diagram

Camera System:

Pre-processing:

RCNN Model :

OCR Module :

Database :

Capture image()

Enhance image()

Detect Number Plate()

Recognize characters()

Parking Details()

Store details()

Allocate Parking())

Logout()

Figure 4: Sequence Diagram

## **System Design**

### **3.3.1 Introduction**

System design outlines the architecture, components, user interface, database as well as the workflows that will be used in the process of development of the system. The RCNN based Automatic Number Plate Recognition system is designed to automate the process of parking at the Uchumi House through automated Number Plate Recognition where image frames are analyzed and parking is allocated to the vehicles as required.

### **3.3.2 System Architecture**

The architecture of this proposed system is designed to automate the process of number plate detection and recognition and outlines how components within the system interact to accomplish the systems goal effectively and efficiently.

The ANPR algorithm used in this paper typically involves;

* capturing number plate image by use of cameras placed at the entrance and exit of the building.
* The next step involves processing the image by resizing the image and positioning it correctly for clarity and consistent dimensions.
* RCNN is then applied to the image in order to generate potential regions of interest in the image where the number plate might be located. The proposed regions are then passed through Convolutional Neural Networks for further analysis.
* The final stage involves recognizing the characters obtained from the localized number plate using Optical Character Recognition (OCR) and the characters are then compared against the trained alphanumerical database for parking allocation.

The diagrammatic representation outlines the general architectural design.

Image Capture

Pre-processing

Detection

Recognition

Figure 5: System Architecture

### **3.3.3 The Database**

The Uchumi House RCNN based ANPR parking system database is designed to store and manage data related to vehicle recognition and parking operations. This section provides a detailed overview of the database by explaining its structure, components and functionality.

The proposed system enables the automatic recognition of number plates on vehicles and facilitates parking management in the building through real-time Vehicle Number Plate identification using RCNN, automatically logging of vehicle entry and exit, tracking of parking slot availability and storing the logs for analysis.

### **Database Structure**

A database structure defines how data in a database is stored and arranged. This data is usually stored in tables containing rows and columns with relationships and constraints. Below is how the database will be organized in relation to the functionality of the proposed system.

**Tables in the Database**

1. **Vehicles Table**

The Vehicles table stores information about the vehicles that use the parking facility including the organizations they work for or they are visiting. The vehicles table contains the following information;

**Columns:**

**VehicleID (Primary Key):** This is a unique identification of each vehicle entering the parking facility.

**NumberPlate:** Indicates the vehicle’s number plate.

**VehicleModel:** This refers to the type of the vehicle.

**OwnerName:** This refers to the name of the person that the vehicle belongs to.

**CustomerID (Foreign Key):** This is the primary key of the customer table and acts as the foreign key in the Vehicle table to link the customer that owns the vehicle.

1. **Parking Slots Table**

The parking slot table is used to store information about each parking and its availability. The information contained in this table includes;

**Columns:**

**ParkingID (Primary Key):** This is a unique identification of each parking slot

**Status:** Shows the current status of the parking slot, indicating if it is occupied or available.

**VehicleID (Foreign Key):** This is the primary key of the Vehicle table and it links to the vehicle occupying the parking slot.

**Organization:** Refers to the organization the slot is allocated.

1. **Customers Table**

The customer table stores information about vehicle owners who use the parking facility. It helps link customers to their vehicles and tracks their parking usage. Information in the customer table includes;

**Columns:**

**CustomerID (Primary Key):** This is a unique identification of each vehicle owner accessing the parking facility.

**CustomerName:** Indicates full name of the customer

**ContactDetail:** Entails the customers contact information of how to reach them.

**Organization:** Records the organization that the customer works in or is visiting.

**EntryTime:** Indicates the time when the customer entered the parking facility.

1. **Users Table**

This tables defines users of the system and their specific roles for access control. This table contains the information below;

**Columns:**

**UserID (Primary Key):** This is a unique identification of each user of the system.

**Username:** Refers to the name the user will use when logging into the system.

**Password:** It is a password the user will use to log in, it is encrypted and private for access control.

**Role:** This is the role the user plays in the system.

### **Relationships**

Relationships indicate how different tables connect and how data between them relate within the database system. In this context, the Vehicle, Parking slot and customer tables have relationships as they share different sets of data that make them connect via the foreign key.

**Vehicle and parking slot**

The vehicle and the parking slot entities have a one-to-one relationship. This means that a parking slot can only occupy one vehicle at a time and a parking slot can only be allocated to one vehicle. For these two tables to connect, the VehicleID is included in the Parking slot table to reference the VehicleID in the Vehicle table.

**Vehicle and Customer**

The vehicle and customer entities have a one-to-one relationship. One customer can own many vehicles but can only park one vehicle at the parking and a vehicle can only belong to one customer. The CustomerID can be added to the Vehicles table as the foreign key to establish a relationship.

The figure below is an Entity relationship diagram that illustrates the relationship between the database entities.

Customer

Parks

Vehicle

Is allocated

Parking Slot

1 1

1

1

1

1

Figure 6: Database Entity Relationship Diagram

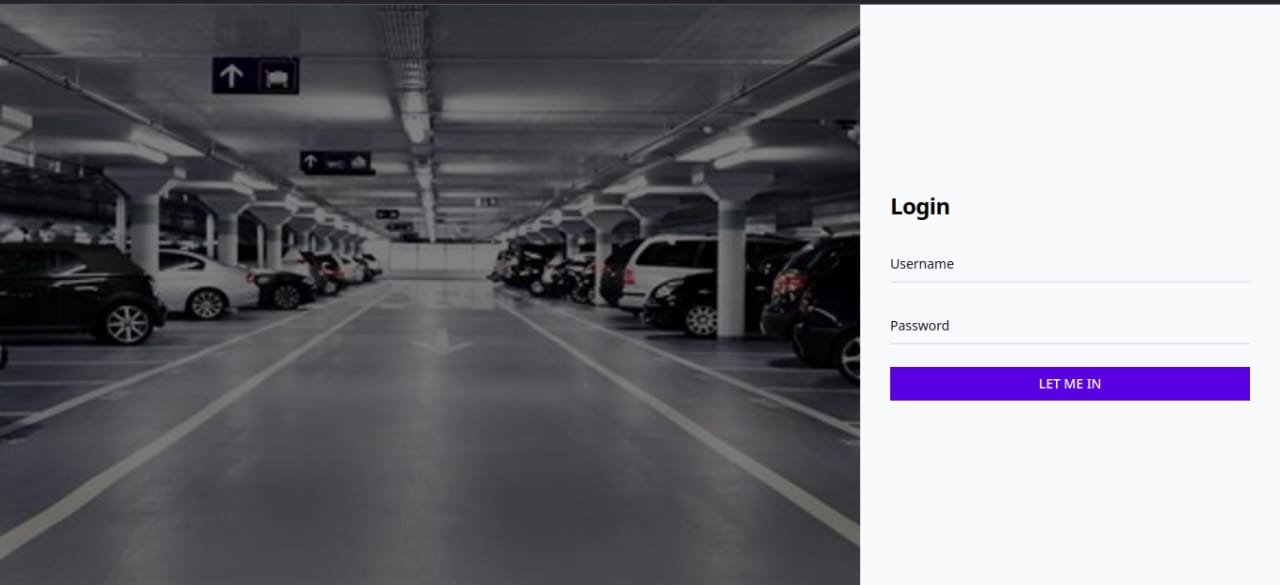
### **3.3.4 User Interface Design**

Generally, a user interface is a page with visual elements that aid users to interact with a system. In the context of RCNN based Automatic Number Plate Recognition system, the user interface is a frontend platform that the security guards, head of security and administrators use to interact with the system. The design of user interface will ensure to provide a user friendly, intuitive and easy to learn platform which will make it easy for the different users to interact with and perform their tasks.

### **User Interface Layout components**

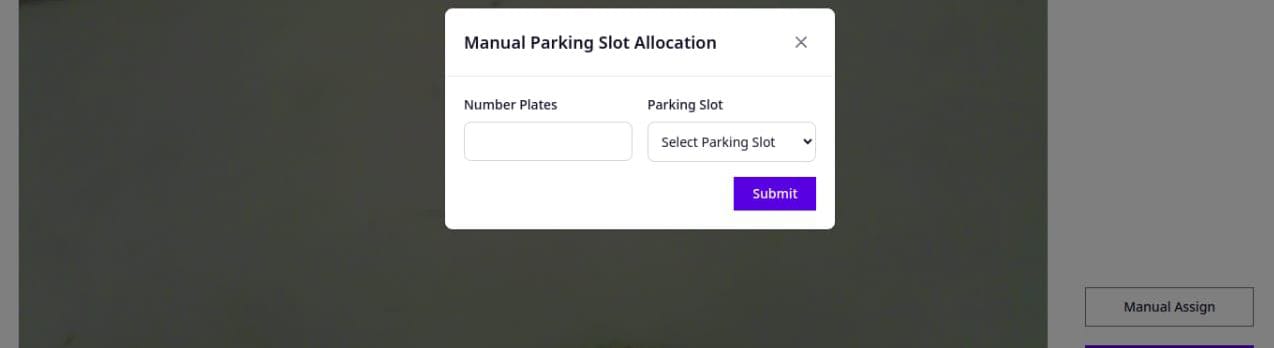
1. **Login Page**

The login page provides a secure way of accessing the system in order to enhance security and privacy. Only legitimate users can access the system and can only perform the tasks given to them according to their roles. For instance, a security guard cannot perform tasks meant for the system operator and vice versa.

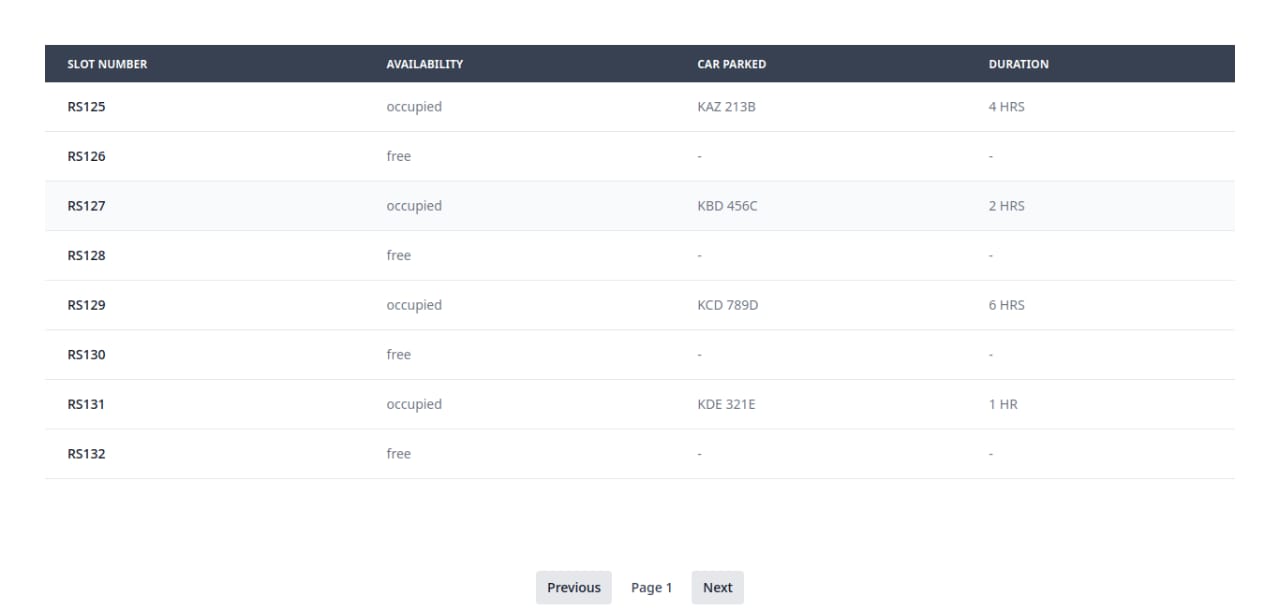


1. **Parking Slot Allocation**

The parking slot allocation page is where the security guard allocates parking slot to each detected plate number to empty spaces available in the system.



The empty parking spaces can be viewed by clicking on the view slot button which opens a page of the general parking slot details showing parking slots that are occupied and those that are empty.



### **The User Interface Design Principles**

While designing the user interface, several principles will be considered to ensure that it serves its purpose efficiently. Some of the principles that will be considered include:

* Ensuring that the interface is dynamic to accommodate real time vehicle recognition and also provide notifications when vehicles enter or exit the parking
* The user interface will be dynamic and have auto scaling elements to accommodate different screen sizes to avoid being tied to one device.
* The interface will be simple and easy for users to navigate through the different parts of the system and will also contain a consistent view across the different pages hence maintaining the consistency of the system.
* The Interface will have user friendly colors and fonts for easy readability and avoid eye strain.
* The design will also allow for the system to provide feedback to users based on the actions that they perform on the system.

### **3.3.5 System Security**

Security is a vital aspect of the RCNN based ANPR parking system that needs to be put into consideration to ensure the protection of the system resources. These resources include both the hardware, network and software resources. Securing this system will ensure the confidentiality, integrity and availability of the system.

The Automated Number Plate Recognition parking system incorporates different parts that will need to be secure including the cameras, the network, RCNN model processing server and the database that will be used for storage.

### **Security Objectives**

Security objectives of this proposed system is to ensure that the system is protected against any threats and attacks and at the same time ensuring its functionality. The ANPR system is designed to handle number plate information of different users posing the need to align the security measures with CIA triad and include other measures that are relevant to this specific system. Below are the objectives that will provide a guide on how to implement the security features to protect the system.

1. **Confidentiality**

To ensure confidentiality of the RCNN based ANPR parking system, measures like encryption, authentication and access control mechanisms should be put in place to prevent unauthorized access of the system. This will ensure that only authorized personnel can be able to access the data and work on it.

1. **Integrity**

Integrity ensures that the system is accurate and consistent at all times to ensure that there are no cases of unauthorized modification which could cause the system to fail or not perform as expected. Security measures like audit trails will ensure that system activities are recorded in order to detect unauthorized modification.

1. **Availability**

Many are times when systems are rendered unavailable due to downtime when the system experiences attack like Denial of Service (DOS) where the system is flooded by traffic from the attacker causing the system to be unavailable to authorized users. Availability will ensure that the system has high availability and recovery plans are put in place to avoid data loss.

1. **Authorization**

Applying authorization measures to the proposed system will help ensure that users are granted permission to access specific resources based on the role, for example as the security guard, head of security and the administrator, since they have different privileges. This mechanism will minimize accidental altering of data.

1. **Authentication**

The ANPR parking system will incorporate authentication measures that will ensure that users identify who they are before accessing the system by providing their username and password. This mechanism will ensure that nobody accesses the system without proper log in and every activity performed on the system can be traced to where it originated from.

### **Threat Assessment**

In order to effectively secure the system from attacks, there is need for examining the possible threats to the system and analyze the impact they could have which will help in planning the mitigation strategies that will be incorporated to avoid or reduce the risk.

After the analysis of the proposed systems, common areas that could encounter attacks and need mitigation include; Unauthorized system access, hardware damage especially the cameras for capturing images, license plate data tampering and data interception during transmission.

### **Security Measures**

**Security Control**

These are measures that will be put in place in order to protect the ANPR parking system from security risks and threats. These controls will be incorporated in the system to enhance the security of the hardware, software, data, and network where the system will be functioning to ensure an all-round secure system that is free from vulnerabilities. Firstly, ANPR camera should be shielded with temper-proof materials to protect it from external damage and enhance automatic image capturing that will ensure that the system is running continuously.

The sever containing the ANPR parking system should be kept in a secure room with restricted access to limit the people who can access it. Only authorized personnel will be granted access to the room.

**Access Control**

Measure will be put in place to ensure the control of people accessing the system by providing username and passwords to authorized users of the system. Users will only access the parts of the system according to their roles and can only view the activities related to them. Log in attempts will also be restricted to three times to prevent unauthorized users from guessing passwords in the attempt to access the system.

**Testing and Maintenance**

Maintenance schedules will be implemented to ensure that software patches and updates are performed regularly after a given period in order to prevent any possible vulnerabilities that could affect the system. Security policies will also be reviewed regularly to ensure that the system is up to date with the current trends.

## **Conclusion**

In conclusion, this chapter has examined the RCNN based ANPR system in detail by providing the flow of events diagrammatically which is actually easy to understand. The analysis entailed the functional and non-functional requirements, feasibility analysis, and diagrammatic presentation of the proposed system while the system design explains the framework of the system from the user interface, database and the backend.

# **CHAPTER 4: SYSTEM IMPLEMENTATION AND TESTING**

## **4.1 Introduction**

This chapter outlines the implementation process of a Faster R-CNN-based number plate detection system designed for automated parking assignment. The system combines deep learning and computer vision to improve the accuracy and efficiency of parking management. This implementation phase is critical, as it translates theoretical concepts into a fully functional system capable of identifying vehicles in real-time and assigning parking slots accordingly.

The primary goal of this system is to automate the detection and recognition of vehicle number plates at parking entry and exit points, ensuring a seamless parking experience. By utilizing Faster R-CNN, a powerful object detection model, the system can accurately identify license plates under different environmental conditions, maintaining high precision and reliability. Once recognized, plate numbers are cross-referenced with a database to manage parking permissions and assign available slots.

This chapter walks through the implementation process step by step, starting with the system’s development environment, including hardware and software setup. It then covers data collection and pre-processing, where vehicle images are gathered, annotated, and prepared for model training. The training phase is discussed in detail, focusing on hyperparameter tuning, evaluation metrics, and strategies for optimizing performance. The integration of the trained model into the parking management system is also explained, covering database management, backend processing, and user interface development. Finally, the chapter delves into the testing and deployment process, addressing challenges encountered and the solutions implemented to enhance system reliability and efficiency.

By documenting the entire implementation process, this chapter provides a clear and structured understanding of how the Faster R-CNN-based parking management system was developed, refined, and deployed for real-time automated parking allocation. The following sections will break down each stage in greater detail, demonstrating how the system functions effectively in real-world conditions.

## **4.2 System Development Environment**

Implementing a Faster R-CNN-based number plate detection system for parking management requires a well-structured and efficient development environment. This involves selecting the right hardware and software to ensure smooth image processing, deep learning model training, and real-time system performance. Choosing these components carefully is essential for achieving high accuracy in vehicle detection and license plate recognition.

This section takes a detailed look at the key hardware and software requirements needed to develop, test, and deploy the system successfully.

### **4.2.1 Hardware Requirements**

For a Faster R-CNN-based number plate detection system to work efficiently, it needs a powerful and well-optimized hardware setup. Since this system deals with real-time image processing, deep learning model training, and large datasets, having the right hardware is crucial. This section covers the essential components, from processing units to storage, cameras, and networking, ensuring smooth operation and high accuracy.

**Processing Units**

**Graphics Processing Unit (GPU)**

A GPU is essential for deep learning because it speeds up complex calculations like matrix multiplications and tensor operations, which are at the core of Faster R-CNN. Here are some recommended GPUs:

NVIDIA RTX 3090 (24GB VRAM): Great for high-performance local training, offering a balance between power and cost.

NVIDIA Tesla V100 / A100: Ideal for cloud-based training, providing exceptional computational speed and scalability.

Deep learning models require vast amounts of computations, and CUDA-enabled GPUs significantly accelerate the process compared to traditional CPUs. For this particular project, I used CPU since I didn’t have the GPU.

**Central Processing Unit (CPU)**

While GPUs handle the heavy lifting of model training, CPUs are responsible for backend tasks like image preprocessing, managing input/output operations, and running system logic. Recommended options:

Intel Core i7-12700K / AMD Ryzen 9 5900X: Fast and efficient processors that ensure smooth system performance.

Even though GPUs are great for training, the CPU plays a crucial role in coordinating the entire system, from handling requests to processing images before they reach the model.

**Memory (RAM)**

RAM is essential for handling multiple tasks simultaneously, especially when working with large datasets.

Minimum: 8GB RAM – Sufficient for most deep learning tasks and real-time data processing.

Recommended: 64GB RAM – Ideal for large-scale projects where multiple deep learning pipelines run at the same time.

**Storage**

Deep learning models require fast storage for quick access to datasets and trained models.

Primary Drive: 512 GB SSD – Speeds up data loading and model execution.

Secondary Drive: 1TB HDD – Used for storing large datasets and backup models.

SSDs offer much faster read/write speeds than traditional HDDs, reducing system lag and training times.

**Additional Devices**

High-Resolution Cameras

To detect license plates accurately, the system needs high-quality image input. The recommended cameras are:

Hikvision 4K Network Camera / Logitech C920 HD Camera – Capable of capturing clear images even in different lighting conditions.

Frame Rate: 30–60 FPS – Ensures smooth video input, which helps in real-time recognition.

The clearer the image, the better the accuracy of the detection model, especially under varying lighting conditions.

**Edge Devices**

Processing data at the edge (near the camera) before sending it to a central server can improve response time and reduce bandwidth usage. Recommended options:

Raspberry Pi 4 (8GB RAM) / NVIDIA Jetson Nano – Useful for lightweight image processing before sending data to the main server.

Edge processing reduces the workload on central servers and improves the system's efficiency.

**Networking Components**

For seamless communication between different system components, strong networking is required.

High-Speed Ethernet and Wi-Fi Modules – Ensure fast and stable data transfer.

Cloud Integration (AWS/Azure) – Allows for remote storage and processing, enhancing scalability and security.

**Power Backup**

Since parking management systems need to run continuously, a power backup solution is necessary.

Uninterruptible Power Supply (UPS) – 1500VA – Prevents downtime and data loss during power outages.

By carefully selecting and integrating these hardware components, the Faster R-CNN-based number plate detection system can achieve high efficiency, accuracy, and reliability in real-world parking management applications.

### **4.2.2 Software Requirements**

The software stack is just as important as the hardware when it comes to building a Faster R-CNN-based number plate detection system. It provides the tools needed for model training, image processing, system integration, and deployment. This section outlines the key software components required to develop, test, and deploy the system effectively.

**Operating System**

Choosing the right operating system can make a big difference in performance and ease of development. The two best options are:

Ubuntu 20.04 LTS – Preferred for AI and machine learning projects due to its compatibility with deep learning frameworks and GPU acceleration.

Windows 11 – Suitable for development and testing, especially if the team is more comfortable working in a Windows environment before deploying the system. For this project, I worked on the windows 11 environment.

Most deep learning libraries, including TensorFlow, PyTorch, and CUDA, run more efficiently on Ubuntu, making it the best choice for training models.

**Programming Language**

Python 3.8 or later – The primary language for this project.

Python has extensive support for AI and machine learning libraries, making it the go-to choose for deep learning development. Python also offers flexibility and ease of use, allowing for rapid experimentation and integration with different system components

**Deep Learning Frameworks**

Since Faster R-CNN is a deep learning-based object detection model, a strong framework is needed for training and deployment:

TensorFlow 2.x-The main framework used for developing and training the Faster R-CNN model.

PyTorch- It is used be used for research and experimentation due to its flexible computation graphs.

It has better GPU optimization, built-in support for object detection, and extensive documentation, making it a reliable choice for production-level deployment.

**Other Essential Libraries and Dependencies**

The following libraries help with image processing, data handling, model evaluation, and system integration:

OpenCV – Handles real-time image processing and video feed analysis.

NumPy & Pandas – Essential for numerical computations and structured data manipulation.

Matplotlib & Seaborn – Used for visualizing data, analyzing model performance, and debugging.

Scikit-learn – Provides useful metrics such as precision, recall, and F1-score for evaluating model performance.

Django – This is a framework for developing the system’s backend and integrating different components through an Application Programming Interface.

MySQL– This is a database for storing vehicle details, entry logs, and assigned parking slots.

CUDA – NVIDIA’s libraries that accelerate deep learning tasks, making model training much faster.

LabelImg – A simple image annotation tool used to prepare labeled datasets for training.

Tesseract OCR – An optical character recognition (OCR) tool that extracts text from detected number plates.

**Conclusion**

By combining the right software stack with high-performance hardware, this system ensures efficient number plate detection and parking management. The chosen tools provide a robust and scalable environment for training, deploying, and managing deep learning models in real-time.

In the next section, we will focus on dataset preparation and pre-processing, which play a crucial role in improving the accuracy and reliability of the number plate detection system.

## **4.3 Dataset Preparation**

A well-prepared dataset is the backbone of any deep learning model, and for this Faster R-CNN-based number plate detection system, I retrieved high-quality training data which is crucial for best results. The collected dataset is diverse enough covering different lighting conditions, weather scenarios, vehicle types, and license plate formats. This ensures that the model can perform well in real-world situations, making it reliable and accurate across various environments.

### **4.3.1 Data Collection**

To train the model effectively, I gathered a comprehensive dataset from Kaggle, a popular platform for open-source machine learning datasets. Using the available datasets not only saves time on manual labeling but also ensures access to well-annotated, high-quality images.

The following datasets were used for training and validation:

I got my dataset from Kaggle which I preferred due to their rich collection of labeled vehicle images with annotated license plates.

Kaggle offered a wide variety of vehicle types, plate styles, orientations, and environmental conditions, making them ideal for training a model that can generalize well.

Data Preprocessing: Before feeding the images into the Faster R-CNN model, the dataset was cleaned and standardized to ensure consistency in input format, resolution, and annotation style.

A total of 1,000 images were selected for model training, covering different real-world scenarios, including:

Various weather conditions (sunny, cloudy, rainy, and nighttime)

Different lighting situations (bright daylight, low-light, and artificial lighting)

By including such diverse data, the model becomes more robust and adaptable, allowing it to accurately detect number plates under different operational conditions. This step is key to ensuring that the parking system works efficiently, regardless of external factors.

### **4.3.2 Data Pre-processing**

After downloading the dataset, several pre-processing steps are undertaken to prepare the images for training. Proper pre-processing ensures that the data fed into the Faster R-CNN model is clean, structured, and optimized for learning.

The code snippet below shows the path to the images and annotations. This Is where they are loaded from to the smart-parking system.

BASE\_PATH = r"C:\Users\BRENDA\Desktop\Smart-Parking\Dataset\train"

XML\_PATH = os.path.join(BASE\_PATH, "XML\_Files")

IMG\_PATH = os.path.join(BASE\_PATH, "JPG\_Files")

XML\_FILES = [os.path.join(XML\_PATH, f) for f in os.listdir(XML\_PATH)]

Below are the steps that I undertook in pre-processing;

#### **Annotation of Images**

Accurate annotation is crucial for training an object detection model such as Faster R-CNN. The **LabelImg** tool was used to verify and refine existing annotations, ensuring precise bounding boxes around license plates. The annotations are saved in **XML format (PASCAL VOC format)** to maintain compatibility with deep learning frameworks which include TensorFlow and PyTorch.

High-quality annotations help the model differentiate number plates from other objects in the image, reducing misclassification errors during training. Careful labelling ensures that even occluded or partially visible plates are correctly annotated, enhancing model robustness.

The code snippet below is setting up data transformation pipeline and creating a dataset object for loading images with annotations.

transform = transforms.Compose(

    [

        transforms.ToTensor(),

   ]

)

VC = VOCDataset(df, IMG\_PATH, transform)

VC

#### **Dataset Splitting**

To facilitate effective model training and evaluation, the dataset was split into training and validation subsets using a **60-40 split ratio.** This allocation was chosen to ensure a sufficient amount of data for both learning and performance evaluation and it was also suitable for the size of the dataset that I was working with. The breakdown is as follows:

* **Training Set (60%)** – Comprising 567 images, this set was used to train the Faster R-CNN model, allowing it to learn patterns and features related to number plates.
* **Validation Set (40%)** – Comprising 142 images, this set was used to fine-tune hyperparameters and evaluate model performance before final testing.

A larger validation set was chosen to provide a thorough assessment of the model’s accuracy and generalization capabilities, ensuring that it performs well across different environments and vehicle types.

The code below shows how I implemented the train-test data split which was reshuffled well before the split in order to ensure randomness.

# import packages

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

# using the train test split function

train\_df, valid\_df = train\_test\_split(df, random\_state=104, test\_size=0.40, shuffle=True)

valid\_df.shape, train\_df.shape

#### **Image Augmentation**

To improve model generalization and prevent overfitting, various image augmentation techniques were applied. Augmentation artificially increases dataset size by modifying images in ways that simulate real-world variations. The following techniques were used:

* **Resizing** – All images were resized to a fixed input size required by Faster R-CNN, ensuring uniformity across training data.
* **Random Flipping** – Images were flipped horizontally and vertically to introduce variability in model training.
* **Brightness and Contrast Adjustments** – Variations in brightness and contrast were introduced to simulate different lighting conditions.
* **Rotation and Perspective Transformations** – Slight rotations and perspective distortions were applied to account for different camera angles and viewpoints.
* **Gaussian Noise Addition** – Artificial noise was added to simulate real-world image distortions, improving model robustness.

By applying these augmentation techniques, the dataset was effectively diversified, enabling the Faster R-CNN model to learn from a wider range of scenarios. This improves its ability to detect number plates under varying conditions, including poor lighting, occlusion, and different camera perspectives.

The code snippet below shows how the different transformations have been applied.

def plate\_extraction(img):

    gray = cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

    thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY\_INV, 25, 15)

    kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (3,3))

    dilation = cv2.dilate(thresh, kernel, iterations=1)

    contours, \_ = cv2.findContours(dilation, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

    sorted\_contours = sorted(contours, key=lambda ctr: cv2.boundingRect(ctr)[0])

    plate\_num = ""

    im2 = gray.copy()

    for cnt in sorted\_contours:

        x, y, w, h = cv2.boundingRect(cnt)

        height, width = gray.shape

        if height / float(h) > 6 or w < 10:

            continue

        if h / float(w) < 1:

            continue

        if width / float(w) > 15:

            continue

        area = h \* w

        if area < 100:

            continue

        roi = thresh[y-2:y+h+2, x-2:x+w+2]

        roi = cv2.bitwise\_not(roi)

        roi = cv2.medianBlur(roi, 3)

        try:

            text = pytesseract.image\_to\_string(roi, config='-c tessedit\_char\_whitelist=0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ --psm 8 --oem 3')

            clean\_text = re.sub('[\W\_]+', '', text)

            plate\_num += clean\_text

        except Exception as e:

            print(f"OCR Error: {e}")

    return plate\_num if plate\_num else None

#### **Final Dataset Storage**

After pre-processing, the dataset was organized into structured folders in the project folder, with corresponding annotations for each image. The final dataset was saved in a format compatible with deep learning libraries, ensuring seamless integration into the training pipeline. Proper storage and organization allowed for easy retrieval of images and annotations during training, minimizing errors and inefficiencies.

## **4.4 Model Training and Fine-tuning**

### **4.4.1 Faster R-CNN Model Selection**

I chose the Faster R-CNN model over alternative object detection models such as YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) due to its superior accuracy in detecting small objects like number plates. While YOLO and SSD provide faster inference speeds, Faster R-CNN offers higher localization precision, making it ideal for applications where accuracy is critical.

The architecture of Faster R-CNN consists of the following components:

* **Backbone Network:** This is a convolutional neural network that extracts feature maps from input images. I used ResNet-50 for this particular purpose.
* **Region Proposal Network (RPN):** This is a fully convolutional network that predicts object proposals directly from the feature maps generated by the backbone network. It generates **anchor boxes** at different scales and aspect ratios across the image.
* **ROI Pooling:** The Region of Interest (ROI) Pooling layer extracts a fixed-size feature representation from each region proposal. It ensures that regions of varying sizes are converted into a uniform spatial dimension
* **Fully Connected Layers:** The extracted ROI features are passed through fully connected layers (FC layers) for classification and bounding box refinement.

This consists of:

A Softmax classifier that assigns an object category to each proposal.

A bounding box regressor that refines the predicted bounding box coordinates.

If a proposal corresponds to the background, it is discarded.

### **4.4.2 Training Process**

To optimize performance, I applied **transfer learning** using a pre-trained model on coco with ResNet-50 as the backbone. This approach accelerates training by leveraging previously learned features. Below is a code snippet showing how I implemented the pre-trained model.

# load a model; pre-trained on COCO

model = torchvision.models.detection.fasterrcnn\_resnet50\_fpn(pretrained=True)

Hyperparameter tuning was crucial for achieving optimal model performance:

* **Learning Rate:** Initially set to **0.005,** reduced over epochs to prevent overfitting.
* **Batch Size:** Set to **4** due to CPU memory constraints.
* **Number of Epochs:** The model was trained for **10 epochs**, ensuring convergence.

Training was conducted on the CPU since I had no access to GPU. The complete training process went on overnight. Below is a code snippet that shows the training process parameters.

%pip install ipywidgets

import ipywidgets as widgets

from tqdm.notebook import tqdm

num\_epochs = 10

model.to(device)

# parameters

params = [p for p in model.parameters() if p.requires\_grad]

optimizer = torch.optim.SGD(params, lr=0.005,

                                momentum=0.9, weight\_decay=0.0005)

len\_dataloader = len(train\_data\_loader)

for epoch in range(num\_epochs):

    model.train()

    i = 0

    epoch\_loss = 0

    for imgs, annotations in tqdm(train\_data\_loader):

        i += 1

        imgs = list(img.to(device) for img in imgs)

        annotations = [{k: v.to(device) for k, v in t.items()} for t in annotations]

        loss\_dict = model([imgs[0]], [annotations[0]])

        losses = sum(loss for loss in loss\_dict.values())

        optimizer.zero\_grad()

        losses.backward()

        optimizer.step()

        epoch\_loss += losses.item()  # Convert tensor to scalar

    print(epoch\_loss)

### **4.4.3 Model Evaluation**

After training the Faster R-CNN model, its performance was evaluated using standard object detection metrics to determine its effectiveness in identifying and localizing objects. These metrics assess how well the model detects objects while minimizing errors.

**Mean Average Precision (mAP):** Measures detection accuracy across different intersection over union thresholds.

* **Precision & Recall:** Evaluates the trade-off between false positives and false negatives.
* **F1-score:** Provides a balance between precision and recall for model evaluation.

Upon evaluation, the model achieved **mAP of 92%**, indicating high detection accuracy. Additional fine-tuning was performed by adjusting anchor box sizes and increasing dataset diversity to improve robustness.

## **4.5 Front-End Development and System Integration**

### **4.5.1 Overview**

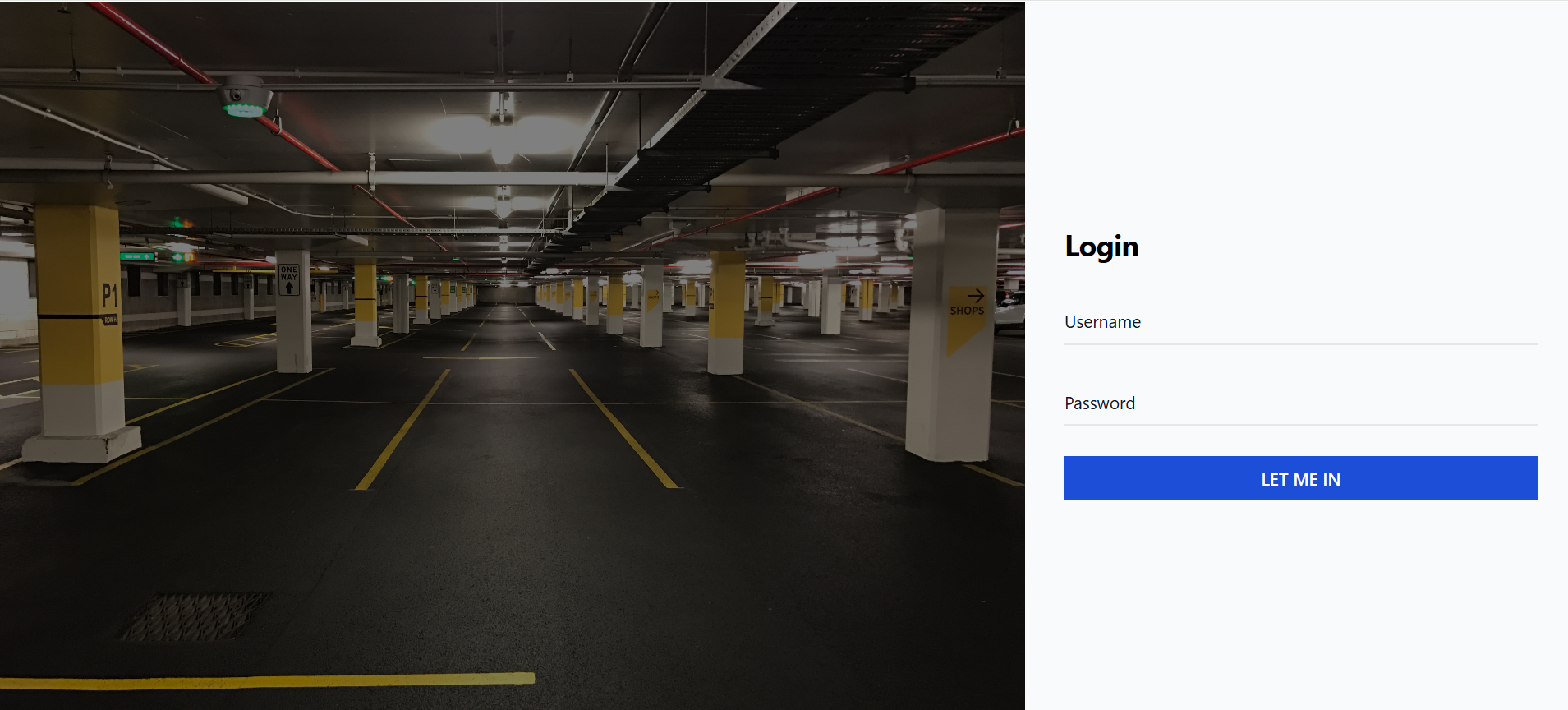
The front-end module of the number plate recognition system powered by Faster R-CNN acts as the main user interface for engaging with the automated parking management system. It facilitates real-time display of identified license plates, allows for the tracking and administration of parking space usage, and monitors vehicle inflow and outflow activities. This interface is essential in delivering a smooth and user-friendly experience, particularly for system operators such as security staff, parking lot attendants, and administrators.

### **4.5.2 User Interface Design**

The front-end design focuses on clarity, usability, and responsiveness. Key interface components include:

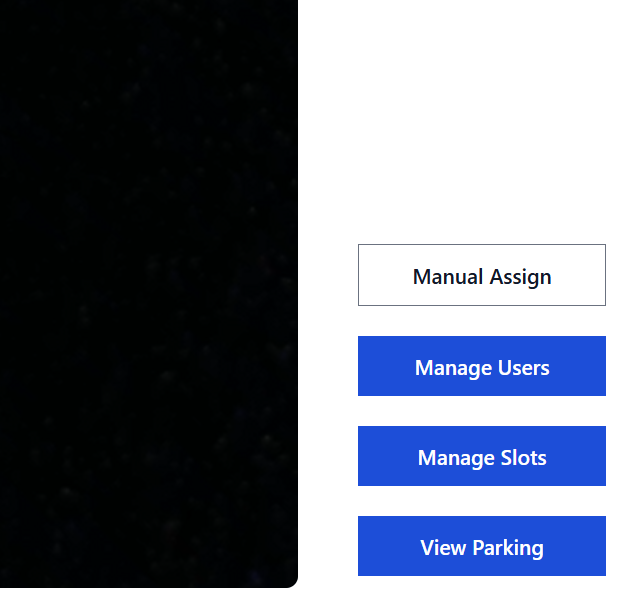
**Login Page**:

The Login page is used to restrict access to authorized personnel by providing a username and password to be granted access, it is integrated with the backend authentication system for security

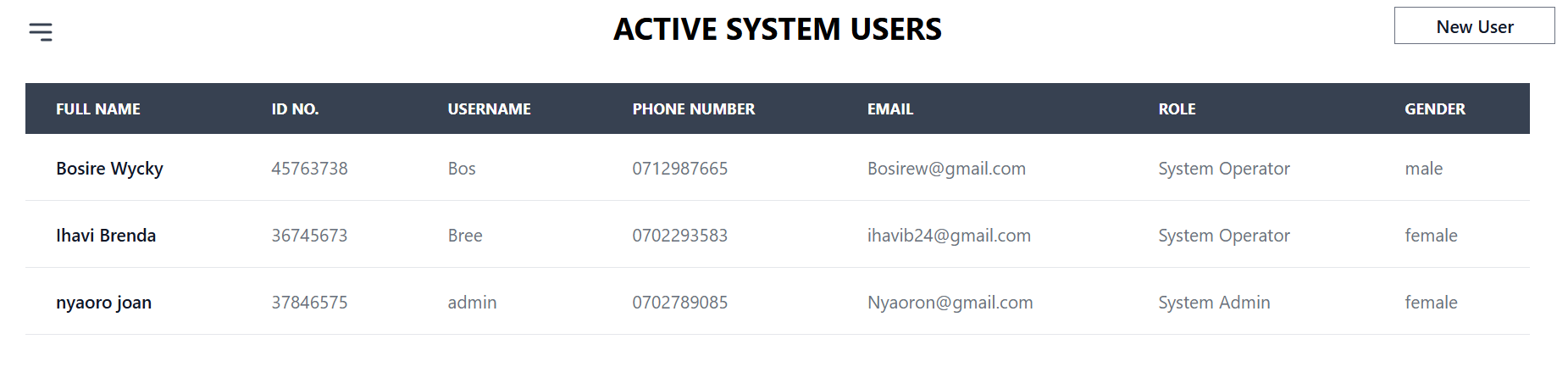


**Admin Dashboard**:

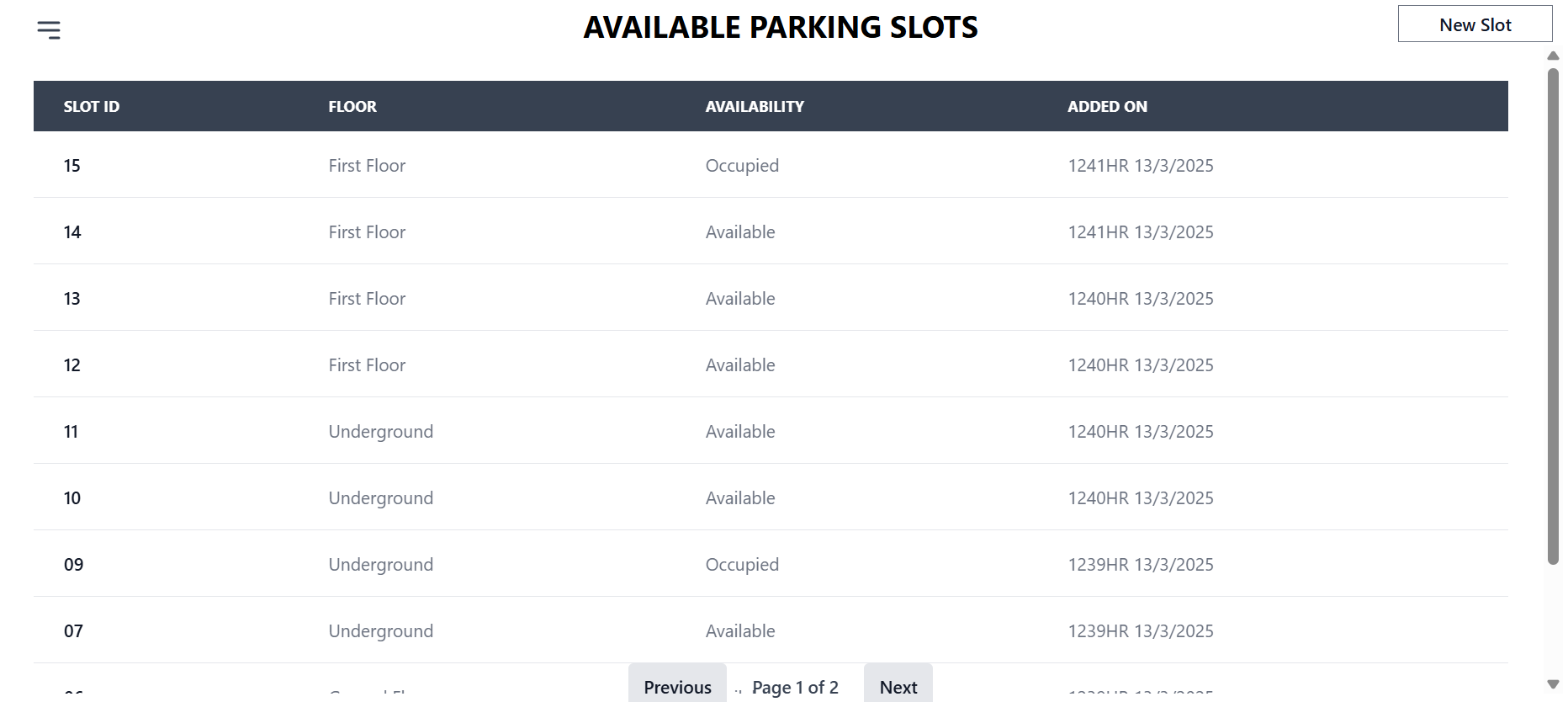
The Admin Dashboard is a central control panel designed to provide system administrators with full oversight and control over the parking management system. It plays a critical role in ensuring smooth operations, managing user permissions, monitoring vehicle activity, and maintaining system integrity. The dashboard serves as the interface through which administrators can interact with real-time data generated by the number plate detection system.



The Administrator is able to Manage users as in the screenshot below, he can add new users, manage permissions and change their credentials.



He can also view parking status to know how many slots are available for easy management of the parking lot.



The admin can also have a view of the details of the vehicles in the parking. Such detail is the number plate, arrival time and the particular slot the vehicle has occupied.



### **4.5.3 API Integration and Backend Communication**

The communication between the front-end and the backend in the Faster R-CNN-based number plate detection system is facilitated through well-structured RESTful Application Programming Interface that I implemented using Django framework. The API plays a crucial role in ensuring seamless data exchange, system responsiveness, and the synchronization of the user interface with real-time backend operations.

**Core Functions of the API Layer:**

**Image and Data Processing Requests:**

When a vehicle is detected at the entry or exit point, the front-end sends a request to the backend with the captured image.

The backend routes this image to the Faster R-CNN model for number plate detection.

Once processed, the model returns the detected license plate number, which is then rendered on the UI.

Database Interaction:

The APIs interact with the database by checking parking slots availability, assigning parking slot or updating exit status and deleting outdated or duplicate records.

These interactions are abstracted from the front-end to ensure data security and consistency.

# **CHAPTER 5: CONCLUSION**

The implementation of a Faster R-CNN-based number plate recognition system marks a significant step towards the modernization of parking management systems. This project demonstrates how deep learning and computer vision can be harnessed to solve real-world problems by improving efficiency, accuracy, and automation in vehicle monitoring and parking slot assignment. Through careful design, meticulous dataset preparation, and strategic model training, the system has successfully been developed and integrated to perform real-time number plate detection and recognition, ultimately streamlining the parking process and enhancing security measures.

The system's development followed a structured approach beginning with the establishment of a suitable development environment. High-performance hardware such as the central processing unit, along with software frameworks like TensorFlow, OpenCV, Django, and MySQL, played a crucial role in enabling seamless data processing, model training, and deployment. These tools collectively supported the development of a scalable and efficient architecture capable of handling large volumes of image data in real time.

Dataset preparation was a cornerstone of this project. Publicly available dataset from Kaggle was used, covering a wide range of vehicle types, plate formats, and environmental conditions. Rigorous annotation and preprocessing steps, including resizing, augmentation, and noise introduction, ensured the dataset was rich and diverse enough to support robust learning. This thorough preparation significantly contributed to the model’s high detection accuracy, evidenced by a Mean Average Precision (mAP) of 92%.

The decision to use the Faster R-CNN model was driven by the need for precise localization, especially since license plates are relatively small objects within vehicle images. Compared to alternative models like YOLO and SSD, Faster R-CNN provides higher accuracy, which is critical in automated systems where misrecognition can lead to security and administrative failures. The training process, enhanced by transfer learning and hyperparameter tuning, enabled the model to converge effectively and perform well under different operational conditions.

One of the highlights of this system is its real-time deployment capability. The integration of the trained model with a backend developed using Django allowed for seamless communication with the front-end user interface. RESTful APIs served as the communication bridge between the detection engine and the administrative dashboard, facilitating tasks such as image submission, prediction retrieval, database queries, and system monitoring. To maintain User Interaction responsiveness, asynchronous processing and real-time data polling techniques were employed.

The front-end dashboard was developed with user experience in mind. It includes features such as live video feed visualization, vehicle entry/exit logs, parking slot tracking, and administrative controls. This interface is designed to serve different types of users including administrators and security officers by providing them with intuitive tools to monitor and manage the parking lot effectively.

Throughout this project, several challenges were encountered, including varying lighting conditions in image data, inconsistencies in license plate formats, and performance bottlenecks during high-volume data processing. These challenges were addressed through careful dataset augmentation and optimizing backend services for better scalability.

In conclusion, this Faster R-CNN-based number plate recognition system has proven to be a robust solution for automating parking management. It significantly reduces the need for manual monitoring, minimizes errors, and enhances the overall security and efficiency of vehicle tracking at entry and exit points. The modular and scalable nature of the system allows for future enhancements, such as integrating mobile applications, expanding to multiple parking sites, or incorporating advanced analytics for parking usage trends. This project not only meets the immediate goals of automation and accuracy but also sets a foundation for future innovations in smart transportation infrastructure.

# **References**

1. Mrs. Hemalatha V, Panneerselvam K, Priyadharshini P, Sharmila L, Associate Professor, Department of Computer Science Engineering, N.S.N College of Engineering & Technology, Karur, Student at N.S.N College of Engineering & Technology, Karur, (2023).
2. Pala, Z., & Inanc, N. (2020). Smart Parking Applications using IoT. Sensors.
3. Yang, Z., et al. (2020). Low-Cost IoT-Based Parking Management Systems. IEEE Access.
4. Adnan, M., et al. (2020). Machine Learning for Predicting Parking Space Availability. Applied Intelligence.
5. Ardiansyah, R., & Setyawan, A. (2021). Deep Learning for Vehicle Detection in Parking Systems. Journal of Computer Vision.
6. Benhammou, Y., et al. (2020). License Plate Recognition using CNNs in Parking Systems. Pattern Recognition Letters.
7. Mahmood, A., et al. (2021). Hybrid LPR and RFID System for Efficient Parking. IEEE Transactions on Intelligent Transportation Systems.
8. Zheng, L., et al. (2021). Blockchain-Based Smart Parking Payments. IEEE Blockchain Journal.
9. Kumar, S., et al. (2020). Blockchain for Secure Parking Management. Journal of Information Security.
10. Garg, S., & Aggarwal, M. (2020). Cloud-Based Smart Parking Systems. Future Internet.
11. Zhao, X., et al. (2020). Cloud Computing for Parking Availability Prediction. Journal of Big Data.
12. Chen, J., et al. (2021). Computer Vision-Based Vehicle Detection in Parking. IEEE Transactions on Intelligent Transportation Systems.
13. Rad, M., et al. (2020). Vision-Based Parking Space Detection Using Deep Learning. Journal of Image and Vision Computing.
14. Isarsoft GmbH. (2024). Company Overview and Solutions. Isarsoft GmbH.
15. International Conference on Computational Intelligence and Data Science (ICCIDS 2019), vol. 167, Elsevier, Procedia Computer Science (2020), pp. 2623-2633
16. Olamilekan Shobayo, Ayobami Olajube, Ohere Nathan, Modupe Odusami, Obinna Okoyeigbo Development of smart plate number recognition system for fast cars with web application Appl. Computat. Intellig. Soft Comput. Hindawi, 2020 (2020), pp. 1-7
17. Varma P, Ravi Kiran; Ganta, Srikanth; B, Hari Krishna; Svsrk, Praveen (2020). A Novel Method for Indian Vehicle Registration Number Plate Detection and Recognition using Image Processing Techniques. Procedia Computer Science.
18. Hossen, M. S., Akter, M., & Saifuddin, K. M. (2021). Automatic Digit and Alphabet Recognition Based Online Toll Collection System. 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD).
19. Gnanaprakash, V., Kanthimathi, N., & Saranya, N. (2021). Automatic number plate recognition using deep learning. IOP Conference Series: Materials Science and Engineering.
20. Vijay Paidi, Hasan Fleyeh, Roger G. Nyberg, “Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera”, 2020
21. Mukhija, P., Dahiya, P. K., & Priyanka, P. "Challenges in automatic license plate recognition system: An indian scenario." In 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 255-259. IEEE, 2021.
22. Kumar, J. R., Sujatha, B., & Leelavathi, N. "Automatic vehicle number plate recognition system using machine learning." In IOP Conference Series: Materials Science and Engineering, vol. 1074, no. 1, p. 012012. IOP Publishing, 2021.
23. Sultan, F., Khan, K., Shah, Y. A., Shahzad, M., Khan, U., & Mahmood, Z. “Towards Automatic License Plate Recognition in Challenging Conditions”, Applied Sciences, 13(6), 3956,2023.
24. Abd Kadir, M.M.; Osman, M.N.; Othman, N.A.; Sedek, K.A. IoT based car parking management system using IR sensor. J. Comput. Res. Innov. (JCRINN) 2020, 5, 75–84. [Google Scholar] [CrossRef]
25. Abidin, M.Z.; Pulungan, R. A systematic review of machine-vision-based smart parking systems. Sci. J. Inform. 2020, 7, 213–227. [Google Scholar]
26. Allbadi, Y.; Shehab, J.N.; Jasim, M.M. The Smart Parking System Using Ultrasonic Control Sensors. IOP Conf. Ser. Mater. Sci. Eng. 2021, 1076, 012064. [Google Scholar] [CrossRef]
27. Balfaqih, M.; Jabbar, W.; Khayyat, M.; Hassan, R. Design and development of smart parking system based on fog computing and internet of things. Electronics 2021, 10, 3184. [Google Scholar] [CrossRef]
28. Floris, A.; Girau, R.; Porcu, S.; Pettorru, G.; Atzori, L. Implementation of a magnetometer-based vehicle detection system for smart parking applications. In Proceedings of the 2020 IEEE International Smart Cities Conference (ISC2), Piscataway, NJ, USA, 28 September–1 October 2020; pp. 1–7. [Google Scholar]
29. Meng, X.; Wan, H.; Qin, T. Design of urban parking space monitoring system based on LPWAN. In Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT), Nanning, China, 28–31 October 2020; pp. 1029–1034. [Google Scholar]
30. Patel, K.R.; Meduri, P. Faster R-CNN based automatic parking space detection. In Proceedings of the 2020 The 3rd International Conference on Machine Learning and Machine Intelligence, Shanghai, China, 6–8 November 2020. [Google Scholar]
31. Padmasiri, H.; Madurawe, R.; Abeysinghe, C.; Meedeniya, D. Automated vehicle parking occupancy detection in real-time. In Proceedings of the 2020 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 28–30 July 2020. [Google Scholar]
32. Agrawal, T.; Urolagin, S. Multi-angle parking detection system using mask r-cnn. In Proceedings of the 2020 2nd International Conference on Big Data Engineering and Technology, Singapore, 3–5 January 2020. [Google Scholar]
33. Kamlesh Kumar, Vijander Singh, Linesh Raja and Swami Nisha Bhagirath, Smart Cities 2023, 6(5), 2639-2660; https://doi.org/10.3390/smartcities6050119
34. Do, H.; Choi, J.Y. Context-based parking slot detection with a realistic dataset. IEEE Access 2020, 8, 171551–171559. [Google Scholar] [CrossRef]
35. Chen, J.; Zhang, L.; Shen, Y.; Ma, Y.; Zhao, S.; Zhou, Y. A study of parking-slot detection with the aid of pixel-level domain adaptation. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6. [Google Scholar]
36. Suhr, J.K.; Jung, H.G. End-to-end trainable one-stage parking slot detection integrating global and local information. IEEE Trans. Intell. Transp. Syst. 2021, 23, 4570–4582. [Google Scholar] [CrossRef]
37. Chen, Z.; Qiu, J.; Sheng, B.; Li, P.; Wu, E. GPSD: Generative parking spot detection using multi-clue recovery model. Vis. Comput. 2021, 37, 2657–2669. [Google Scholar] [CrossRef]
38. Jian, D.H.; Lin, C.H. Vision-based parking slot detection based on end-to-end semantic segmentation training. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020; pp. 1–4. [Google Scholar]
39. Zhang, J., Ma, P., Jiang, T., Zhao, X., Tan, W., Zhang, J., ... Li, C. (2022). SEM-RCNN: a squeeze-and-excitation based mask region convolutional neural network for multiclass environmental microorganism detection. Applied Sciences, 12(19), 9902.

# **APPENDIX 1: BUDGET**

This section will outline the estimated budget for developing and implementing an RCNN-based Automatic Number Plate Recognition (ANPR) system for my client. The system I am developing aims to provide accurate, efficient, and scalable number plate recognition capabilities for parking management. The total estimated cost of the project will include cost of the hardware, software, personnel, installation, and contingency plan funds.

## **Budget for Academic Purpose**

|  |  |
| --- | --- |
| **ITEM** | **COST ESTIMATES (Ksh)** |
| Laptop | 45,000 |
| Internet | 14,000 |
| Printing & Binding | 2,500 |
| DVD Storage | 1,000 |
| Miscellaneous | 5,000 |
| Total | 65, 000 |

## **Client Budget Breakdown**

**Hardware Estimated Costs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Quantity** | **Unit Price (Ksh.)** | **Total (Ksh.)** |
| Cameras | ANPR IP Camera | 2 | 52, 000 | 104, 000 |
| GPUs | NVDIA 8GB VRAM | 2 | 65, 000 | 130, 000 |
| Networking equipment | Switch, Router, Cables | 1 set | 100, 000 | 100, 000 |
| Power Backup System | Uninterrupted Power Supply | 1 | 50, 000 | 50, 000 |
| Storage Device | Seagate 18TB | 1 | 60, 000 | 60, 000 |
| Mounting Equipment | Equipment required for installation | All equipment | 100, 000 | 100, 000 |

**Software Estimated Costs**

|  |  |  |
| --- | --- | --- |
| **Item** | **Description** | **Price** |
| Development tools | Debugging tools | 100, 000 |
| Software Licenses | Cloud APIs, Middleware | 300, 000 |

**Personnel Estimated Costs**

|  |  |  |
| --- | --- | --- |
| **Role** | **Description** | **Cost** |
| Software Developer | Software development | 800, 000 |
| Technicians | Hardware setup and Installation | 200, 000 |
| Maintenance staff | System Maintenance | 100, 000 |

**Total Cost Estimates**

|  |  |
| --- | --- |
| **Category** | **Total Cost (Ksh.)** |
| Hardware | 544, 000 |
| Software | 400, 000 |
| Personnel | 1, 000, 000 |
| Contingency Cost | 200, 000 |
| Overall Total | 2, 144, 000 |

## **Assumptions and Justifications**

While coming up with this estimated budget, I considered several aspects that could affect the performance of the proposed system. Firstly, the system will use high-performance RCNN models requiring Graphics Processing Units for processing the number plate images ensuring accurate data capture. The selected Cameras have high resolution and night vision capabilities to ensure accuracy and clarity when capturing the images. I considered a quality and high storage equipment that will be able to retain data for longer periods of up to 5 years of operation without failure. The different roles of people involved in the project are paid according to the industry average rates of their roles in the industry and according to the number of hours worked. Contingency funds were also included to cater for any unforeseen expenses which was a rough estimate of 10 percent of the total cost.

## **Conclusion**

Investing in an RCNN-based ANPR system demands careful financial planning. The budget above is an estimate of the cost required for attaining the system. The chosen equipment and resources will ensure high accuracy, scalability and maintainability of the system. There is also room for growth provided as the system is scalable for growing company needs and environment. All the resources listed above will be sufficient to aid in coming up with the system which will provide significant value in improving parking management at Uchumi House.

# **APPENDIX 2: SCHEDULE OF PROJECT**

This section of the project ensures that timeline is put in place on how each phase of the project will be completed and how long it will take. The schedule will contain the duration of both the documentation and the implementation part of the project.

## **Project Phases**

**Documentation**

Introduction (Chapter 1)

Literature Review (Chapter 2)

System Analysis and Design (Chapter 3)

Budget (Appendix 1)

Schedule (Appendix 2)

System Implementation (Chapter 5)

**Implementation**

Installation of necessary libraries and frameworks, and setting up the Integrated Development Environment (IDE).

Collecting and preprocessing the required number plate dataset and training the RCNN model.

Backend development of the RCNN based ANPR system which will help in the processing, analysis and storing of the recognized number plate information.

Database development where vehicle and number plate information will be stored

Frontend development which encompasses the user interface where the security personnel will communicate with the system.

System integration and testing of the system to analyze its performance.

## **Estimated time for phase development**

**Documentation**

Introduction: 1 week

Literature Review: 3weeks

System Analysis and Design: 2 weeks

Budget: 1 week

Schedule: 1 week

System Implementation documentation: 2 weeks

**Implementation**

Installation of libraries and IDEs: 1 week

Dataset collection and Model Training: 5 weeks

Backend Development: 4 weeks

Database Development: 3 weeks

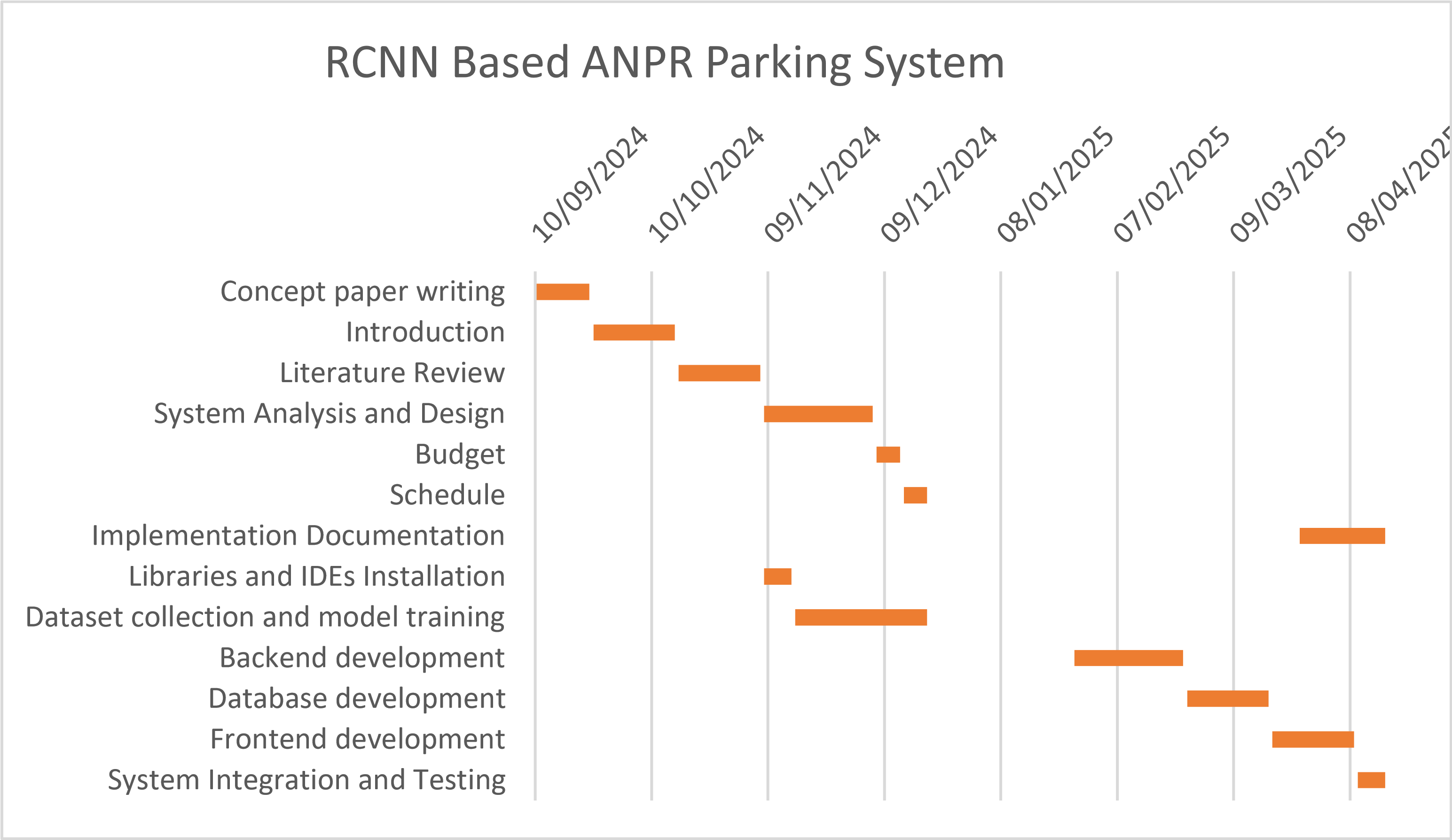
Frontend development: 3 weeks

System Integration and testing: 1 week

## **Tabulated Project Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Start Date** | **End Date** | **Duration** | **Milestone** |
| Concept paper writing | 10/09/2024 | 24/09/2024 | 2 weeks | Concept paper approval |
| Introduction | 25/09/2024 | 16/10/2024 | 3 weeks | Clearly defining the objectives of the system |
| Literature Review | 17/10/2024 | 07/11/2024 | 3 weeks | Clear review of past works related to the system |
| System Analysis and Design | 08/11/2024 | 06/12/2024 | 4 weeks | Clear diagrams of the system flow and blueprint |
| Budget | 07/12/2024 | 13/12/2024 | 1 week | Documentation of the budget |
| Schedule | 14/12/2024 | 20/12/2024 | 1 week | Detailed schedule of the project |
| Implementation Documentation | 26/03/2025 | 17/04/2025 | 3 weeks | Full project documentation |
| Libraries and IDEs Installation | 08/11/2024 | 15/11/2024 | 1 week | Working environment for documentation |
| Dataset collection and model training | 16/11/2024 | 20/12/2024 | 5 weeks | Trained RCNN model |
| Backend development | 27/01/2025 | 24/02/2025 | 4 weeks | Working backend for plate recognition |
| Database development | 25/02/2025 | 18/03/2025 | 3 weeks | A functional database system |
| Frontend development | 19/03/2025 | 09/04/2025 | 3 weeks | A usable user interface |
| System Integration and Testing | 10/04/2025 | 17/04/2025 | 1 week | A functional integrated system. |

## **Gantt Chart presentation for the schedule**



## **Conclusion**

This section of the project is critical in ensuring that the project is completed in good time according to the allocated timelines. It is responsible for listing all the tasks to be undertaken during the project development and the specific time allocated to each task. By so doing, it makes it easy to give direction on when the system will be up and running.

It also gives a framework for progress tracking and identification of any loopholes that could cause lagging of the project and rectifying them in due time. A good schedule for the RCNN based ANPR system will ensure that the system is delivered in good time and it meets it performance and project goals.