Intelligent Parking System Using Machine Learning

by

Md. Ishtiak Abrar
17201155
Shawon Saha
18101531
Hamim Shabbir Halim
18101542
Shoaib Ahamed Shafi
18101544

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Ishtiak Abrar
17201155

Shawon Saha
18101531

Hamim Shabbir Halim
18101542

Shoaib Ahamed Shafi
18101544

Approval

The thesis titled "Intelligent Parking System Using Machine Learning" submitted by

1. Md. Ishtiak Abrar (17201155)
2. Shawon Saha (18101531)
3. Hamim Shabbir Halim (18101542)
4. Shoaib Ahamed Shafi (18101544)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 25, 2022.

Examining Committee:

Supervisor:
(Member)

Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Md Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

____

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the last two decades, the quantity of automobiles has increased dramatically. As a result, utilizing technology effectively to promote convenient parking at public and private locations becomes critical. Conventional parking schemes make it difficult for vehicles to discover available parking spaces. These methods overlook the fact that vehicles are parked on roadways, poor time management during peak hours, and incorrect vehicle parking in a parking space. Furthermore, typical methods in a parking zone need greater human interaction. There is an urgent necessity to create smart parking systems to address the aforementioned challenges. In order to solve parking management in real time and uncertainty, the authors suggest a smart parking system that makes use of IoT and machine learning techniques. The cloud, cameras, and a cyber-physical system are all used in the suggested approach. The creation of a graphical user experience for managers and end-users is a significant task since it necessitates assuring the parking system's smooth monitoring, management, and security. Furthermore, it must build seamless coordination with a user. The proposed system is effective at wisely dealing with challenges. For instance, it denotes the condition of a parking space to the end-user well beforehand; use of limited and unreserved parking places; incorrect parking; unpermitted parking; proper data analysis of unrestricted and occupied spaces; identifying numerous items in a parking space; fault identification in one or more subsystems; and peak-hour traffic management. The approach saves a lot of time, money, and energy by reducing the need for human involvement.

Keywords: Vehicle Parking System; Machine Learning; Comparative Analysis; YOLOv5; R-CNN

Acknowledgement

First and foremost, thanks to Almighty God, with whose help we were able to finish writing our thesis without too many setbacks.

Second, we would like to thank our supervisor Mr. Moin Mostakim and our co-supervisor Md. Tanzim Reza for their kind assistance and suggestions. They came to our help anytime we needed it.

And lastly, our parents, without whom it could not have been possible. We are now only a few steps away from our graduation thanks to their wonderful prayers and support.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

APS Automated Parking System

AWS Amazon Web Services

$BRTA$ Bangladesh Road Transport Authority

CNN Convolutional neural network

CSP Cloud Solution Provider

CV Computer Vision

DNS Domain Name System

DSI Display Serial Interface

FN False Negative

FP False Positive

FPC Flexible Printed Circuit

$GPIO$ General-Purpose Input/Output

$HTTPS$ Hypertext Transfer Protocol Secure

$IaaS$ Infrastructure as a Service

IOU Intersection over Union

JWT JSON Web Token

LAN Local Area Network

LCD Liquid Crystal Display

$MERN$ MongoDB, ExpressJS, ReactJS, NodeJS

ML Machine Learning

$PaaS$ Platform as a Service

$R-CNN$ Region-based Convolutional Neural Network

$RFID$ Radio-frequency identification

$Saas$ Software as a Service

SHA Secure Hash Algorithms

$SPAC$ Smart Parking Allocation Center

SPS Street Parking System

TLS Transport Layer Security

TN True Negative

TP True Positive

$VODS$ Vehicle Occupancy Detection System

$YOLO$ You Only Look Once

Chapter 1

Introduction

1.1 Thoughts Behind Parking System

Over the last couple of years, the number of automobiles has increased dramatically. The irresponsible parking of automobiles on the roadway generates congestion and pollution. Receiving real-time information regarding parking spots and managing their parking are become more and more challenging for drivers. Thus, a parking system is essential. The deployment of the parking system requires a complex infrastructure, which is costly and impracticable in highly populated and developing countries. Considering historical study, it is evident that no parking system design suited for developing nations has been presented .

1.2 History of Parking System

Paris, France, created an automated parking system in 1905. It resembled a Ferris wheel and had space for eight automobiles. In the 1920s, the paternoster construction became popular because it could stop eight cars in the same amount of ground area as a regular two-vehicle APS. The paternoster was simple to use in a variety of environments, including inside buildings, due to its mechanical simplicity and hint of elegance. Concurrently, Kent Programmed Carports started providing APS with capacities of more than 1,000 automobiles. The Auto Stacker, which debuted in Woolwich, Southeast London, in 1961, was difficult to use. In the United States, interest in APS was revived in the 1990s, and by 2012, there were 25 large currents and cooperative APS initiatives (representing about 6,000 stopping spots). Japan had 1.6 million APS parking spaces in 2012, according to statistics [18].

1.3 Motives and Objectives

The main goal of our project is to create a machine learning (ML)-based parking system component that will help create smart cities. Historical data is utilized as an input by machine learning algorithms to predict future output values. We shall learn about Machine Learning (ML) domains and the importance of this scientific field through this research. The main goals of the research are listed below:

- To deeply understand Machine learning, and how it works.
- Automate traditional parking management with the use of ML.
- Reduce manpower.
- Reduce cost by eliminating unnecessary sensors.
- Utilizing OpenCV with Machine Learning Algorithm.
- Simplify the complexity by using camera sensors.
- Centralize the entire management system by the cloud server.
- To develop a smart parking system for avoiding traffic jams and to maintain safety.
- To demonstrate the significance of Machine Learning (ML) in the actual world and to comprehend the working method and capabilities of Machine Learning (ML).

1.4 Problem Statement

As the world's technology continues to advance, the problems are likewise expanding in size and complexity. Currently, individuals are required to travel vast distances in a short amount of time, which causes a tremendous demand for automobiles. Also, the safety of vehicles is a serious matter of concern for the users. Everyone desires a safe and secure parking spot in their subconscious mind which is a major concern of our research. The first true automobile was made by Karl Benz, from Germany, in 1885/1886 according to the Library of Congress [12]. This step was taken to reduce human effort and maximize the use of time by traveling faster. The number of vehicles has increased traffic jam problems, road accidents, getting late for work, etc. problems are now our regular concerns. Given that the number of automobiles is rising daily, problems on the road are also increasing rapidly. With the vast number of vehicles, people now get in trouble while going outside with their vehicles. Major concerns of the people include reaching a place in time, security of their vehicle while they are not around, finding a good place to rest their vehicle, and so on [2] which leads us to think about how to solve that problem. According to BRTA, Bangladesh has 4471625 vehicles registered up to June 2020 [12]. This vast number of vehicles creates a concern about efficient smart parking systems which is a must for today's world to reduce the complexity of our daily life.

According to [9] Wards Intelligence, the worldwide vehicle population reached 1.32 billion vehicles and trucks in 2016. When compared to 20 years ago, this figure is more than doubled. Furthermore, automobile manufacturing and sales show no indications of slowing down. By 2035, about 2 billion automobiles are predicted to be on the road, with some speculating that this might happen even sooner. The parking environment must change due to the quickly rising number of automobiles on the road and the shortage of parking spaces.

People's poor parking practices contribute to traffic congestion and accidents on the road, which may occasionally result in fatalities. According to the author of [13], "cruising for parking" is the practice of looking for an on-street or off-street lot when a driver gets at the location since there often isn't a secure parking place accessible there.

The parking management issue may be seen from a variety of perspectives. Limited parking spaces, unclear locations of parking spaces, uncertainty about the amount of space in parking spaces, and a tendency to illegally park on streets are all problems. Dealing with the aforementioned issue may get increasingly difficult over the next few years. So to reduce the above-mentioned problems and make everyday life better this research will help us by exploring machine learning which can create a long-lasting positive impact.

1.5 Our Contributions

The earlier research does employ sophisticated technologies. But the equipment wasn't helpful in underdeveloped countries. Implementing an RFID project, for instance, is not practical in Bangladesh, where a large portion of the population cannot read English. Also, using so many sensors is very expensive to implement and to maintain regularly. Furthermore, employing high-tech detecting techniques is difficult to execute and more expensive for every car. We suggest an ML-based architecture that can be implemented with the least amount of resources possible because we are aware that developing nations typically spend little money on research and development. Based on our research, we make the following additions to this study.

In relation to the smart parking system, we give a comparative analysis of ML-based methods.

For future researchers' ease of exploration, we identify the advantages and disadvantages of the various ML-based parking systems.

We propose a theoretical concept for a very cost-effective and realizable intelligent parking system for developing nations that is based on machine learning.

Chapter 2

Related Work

2.1 Background

The first mathematical model of artificial neural networks was published in 1943 in ”A logical calculus of the concepts inherent in nerve activity” by Walter Pitts and Warren McCulloch. This publication marks the beginning of the history of machine learning. The Organization of Behavior by Donald Hebb was released the next year, in 1949. One of the pillars of machine learning progress would be the book’s observations on how behavior relates to neural networks and brain activity [8].

A key tool for reaching the goal of using AI-related technologies is machine learning (ML). Despite being a subset of AI, ML is usually referred to as AI due to its capacity for learning and decision-making. The development of AI [8] included it up to the late 1970s. Then it split out to develop independently. For a variety of cutting-edge technologies, like cloud computing and e-commerce, machine learning has emerged as a critical response tool.

Work on machine learning evolved from a knowledge-driven to a data-driven approach in the 1990s. Scientists began developing computer systems to analyze massive volumes of data and generate conclusions or ”learn” –based on the results. In 1997, IBM’s Deep Blue stunned the world by defeating the chess world champion.

Models of machine learning have grown highly adaptable at continually learning, which increases their precision as they work longer. Combining machine learning methods with modern computer technology increases scalability‘ and efficiency. In conjunction with business analytics, machine learning is capable of resolving a number of organizational issues. Modern ML models may be used to forecast everything from disease outbreaks to stock market fluctuations.

2.2 Image Processing

Image processing is a method for improving pictures and extracting information from them. In contrast to other forms of signal processing, picture processing uses an image as its input and produces either the image itself or a collection of associated qualities or features. The comprehensive process of converting a picture to a digital format, then performing operations to improve the image and extract usable data, is known as image processing.

Different methods are used to images in order to enhance them or extract crucial information. The modification of a two-dimensional image by a computer is known as image processing in mathematics; this is defined as an image written as a function of two real variables, such as the function t(x,y), with a magnitude equal to the intensity of an image at a certain point of coordinates (a, b).

When an image is processed, it may produce a single picture or a group of associated traits or features. Two categories may be made out of this: digital image processing and analog image processing, respectively [18]. Physical copies, such as prints and photographs, may be created using analog image processing techniques like printing and photography. When applying these visual approaches and procedures, image analysts rely on a number of interpretative pillars. Digital images may be edited utilizing computers and modern image processing methods [18].

In order to extract information from digital data, all types of data must undergo three fundamental phases: pre-processing, augmentation, and display, in addition to information extraction (if utilizing digital techniques). These are the three primary phases that all types of data must pass through while being processed digitally [14]. If one wishes for others to repeat and confirm one's results, one must execute image processing appropriately. This procedure entails documenting and reporting processing steps and applying comparable treatments to relevant control pictures to establish comparability [18].

Image processing generally involves three steps: capture, alteration, and presentation. To import the picture, image acquisition tools are required. The outcome of an image analysis and modification may be a changed picture or a report based on the analysis of images [18]. There are several applications for digital image processing in various domains, including color processing, transmission and encoding, microscopy, machine/robot vision, and remote sensing, to name a few. The majority of image processing approaches see an image as a two-dimensional signal that may be processed using typical signal processing techniques.

This operation has three distinct types of goals. Starting with image processing, where the input is a picture and the output is another picture, followed by image analysis, in which the input is a picture and the output is a measurement or dimension, and concluding with computer vision, which translates photos into text. Image comprehension is the process of sending an image to a computer and receiving a standard description of what was seen [18].

Image processing is applied in several scientific and technological applications, including artificial intelligence and remote sensing. Possible applications include face recognition and prediction, item identification, fingerprint detection, sorting, argument reality, microscopic imaging, the mechanism for lane departure alerts, nonphotorealistic representation, diagnostic imaging, and structural imaging. Image processing is now one of the technologies seeing the most rapid growth. It is also an important topic of research in the domains of computer science and engineering among others [18].

Chapter 3

Literature Review

The world is growing so fast with the help of technology nowadays. The number of automobiles worldwide has considerably increased during the last few years. Eventually creating a problem with parking areas for those cars. According to research by the University of Seville [15], parking lots can be detected efficiently with the help of machine learning and image processing algorithms to classify image content. Moreover, for a better view and allocation of the parking lot, the author proposed to use three-dimensional image processing to make the result more accurate. Furthermore, depending on how lots are processed, monitoring approaches can be classified into two categories:

- Counting incoming automobiles to estimate the occupancy of a whole parking lot.
- Inspect each cell for the existence of a vehicle. The majority of vision-based techniques necessitate the presence of cars in specific parking lots.

The authors of [2] used Smart Parking Allocation Center (SPAC), connected through the Driver Request Processing Center. The benefit of SPAC is that it would reduce human resources, which will increase overall efficiency. However, they created a smartphone application enabling two-way communication between Infrastructure to Vehicle and Vehicle to Infrastructure. This system could request parking or reservation. Though it is a beneficial feature, in our country, cars are generally driven by a driver who usually does not use a smartphone or is not literate enough to operate it. Thus, we must implement a more user-friendly approach according to our country's perspective.

The authors of [3] used a wireless sensor network (Motes) to detect occupied car parking slots and notify the server. The cost of implementation for this type of parking management system could rise exponentially. It is tough to set up an electronic device to detect an individual vehicle—a big hassle to maintain and costly. Instead of using Motes, we will try to solve this problem using Vehicle Occupancy Detection System (VODS). For this system to determine if a car has been parked in its designated spot or not, OpenCV and machine learning techniques will be used. Thus we could minimize the cost of placing sensors for individual slots by placing a single camera on top.

The author of [4] used an IR sensor to check the availability of the slot. Which is very cheap to buy compared to [3], but the cost and effort of setting up each IR sensor for each slot with a microcontroller are complex and costly to maintain. Moreover, nowadays, they use an RFID reader for identification and authentication. We could also replace this authentication process with a single camera that will capture the vehicle's number plate and send it to the cloud server. Afterward, with the use of machine learning and image processing, it will extract the vehicle number with a timestamp into the database.

The proposed SPS by [5] includes of an LCD board, a router, a base station, end devices, and a computer server. The final product is made up of a magnetic sensor that can find automobiles. This kind of solution is comparable to [4], where each vehicle need its own magnetic sensor to be set up. It also has some drawbacks like wrong parking or false detection. These issues could be fixed by using machine learning and OpenCV.

In this paper, [1] provides a recommendation system to assist knowledge workers in locating valuable new material. The system [1] creates individualized user models based on activity on company network file servers. This method can cut the time it takes to classify a file by up to 23 times without compromising accuracy.

A parking detection system that uses real-time image processing from video cameras was created by the authors of [11]. The system's accuracy is sufficient (81%). An image capturing module, an image pre-processing module, and an image detecting module make up the system's three parts. The picture is filtered in the first. The detecting module is based on utilizing an unaltered reference picture of a vacant parking lot. A comparison system is constructed using the following images using this grayscale image. The method compares the edges of the most recent picture obtained to the reference image. Finally, a function is used to check for similarities between the comparison and reference pictures' attributes.

The Data privacy system in research has a cloud-based architecture [16]. All of the data for the [17] research is stored on a firebase server. The cloud server is loaded with information on the research [18]. Since the user's location is sensitive information, both the client and server sides must utilize encryption. The HTTPS connection between the cloud application and the client application. The research [16] makes use of a cloud-based approach for mistake correction. In this case, the user has access to a cloud database that is admin-based. The user also has access to the cloud. An administration system for database management is used in the research paper [18]. The administrator in this case has access to the cloud server.

In line with [10], According to the company Parking Ya!, which specializes in the selling of garage spaces, more than 25% of automobiles travelling through cities are looking for a parking spot. The issue may be resolved by using intelligent technology to assist with this task, which will boost operational effectiveness, streamline urban traffic flow, and provide drivers a more enjoyable and time-saving experience. Because fewer cars on the road result in less greenhouse gas emissions, it also lessens the detrimental effects of traffic congestion.

Chapter 4

Approach

4.1 Used Architectures

R-CNN and YOLOv5 are two of the finest algorithms that have been demonstrated to satisfy the problem statement after reviewing all of the studies cited above.

4.1.1 YOLOv5

YOLOv5 is an abbreviation that stands for "You Only Look Once." Which is an algorithm that searches for various objects and recognizes in an image/video. YOLOv5 performs attribute identification as a prediction model and reports the classifier of the recognized photos.

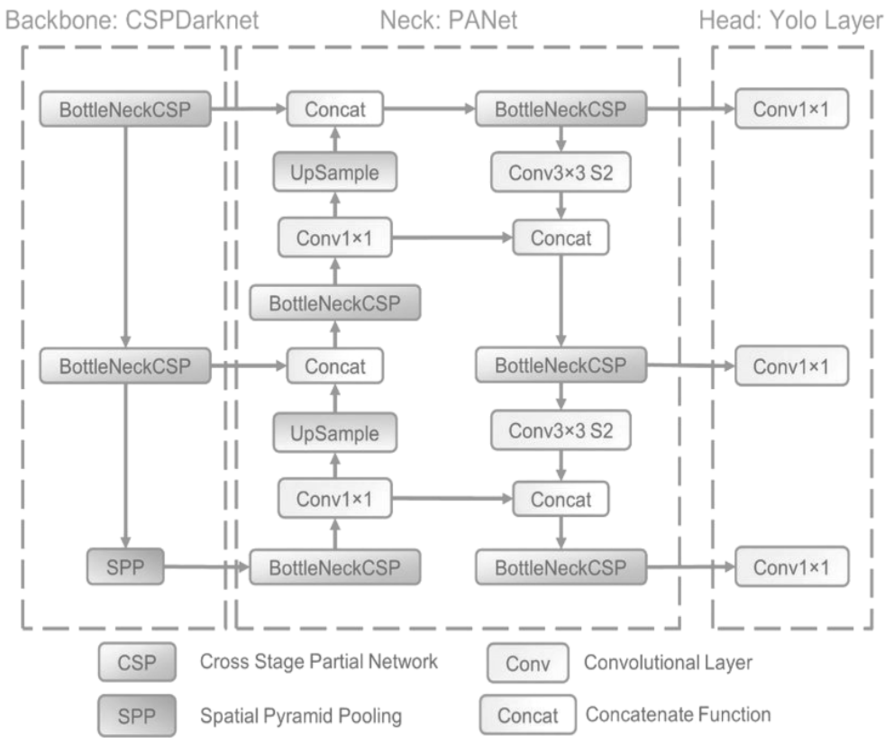


Figure 4.1: Three aspects to YOLOv5's network architecture

The YOLOv5 approach identifies objects in real-time using CNN. As suggested by the name, the method recognizes objects via a single forward propagation over a neural network.

The YOLOv5 method is capable of real-time prediction and has powerful learning capabilities, allowing it to learn the representations of objects and apply them for object recognition. The YOLOv5 technique of prediction delivers consistent results with low background noise. It also boosts the speed of detection since it can predict an object in real-time.

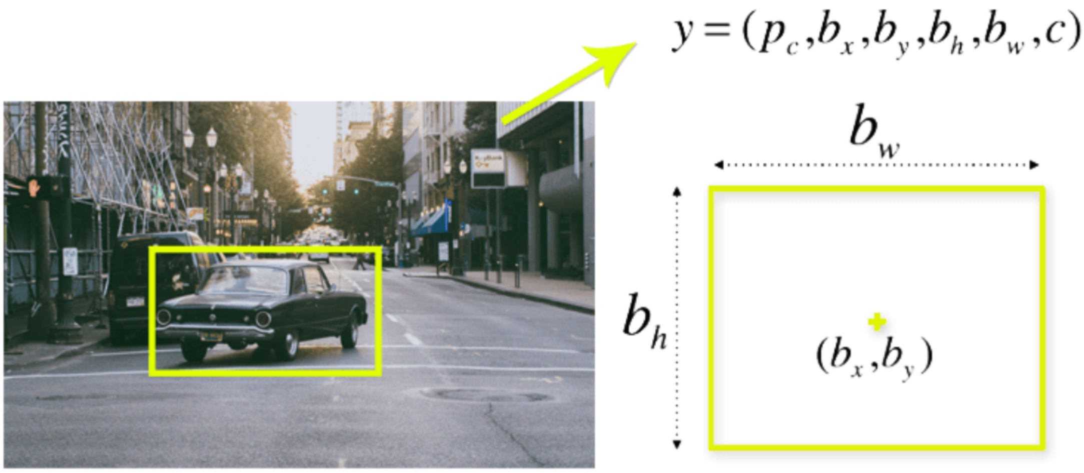


Figure 4.2: YOLOv5

YOLOv5 employs a single bounding box regression to predict object height, breadth, center, and type. The preceding illustration depicts the possibility that an object will appear within the bounding box. According to the standpoint of our country.

4.1.2 R-CNN

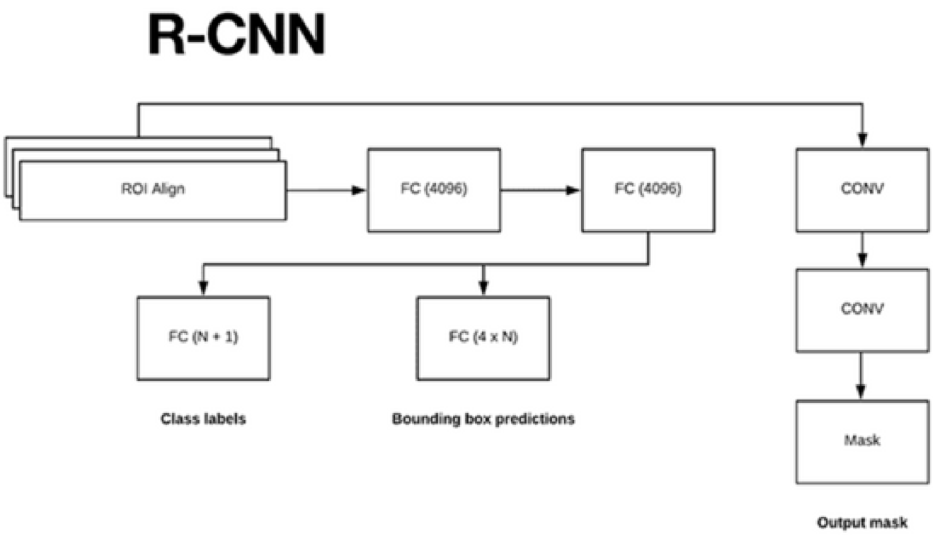


Figure 4.3: R-CNN Architecture

In object detection, classification and localization are two distinct challenges. R-CNN is an acronym for Region-based Convolutional Neural Networks. The series R-CNN is built on regional concepts. Region recommendations are used to find objects inside an image.

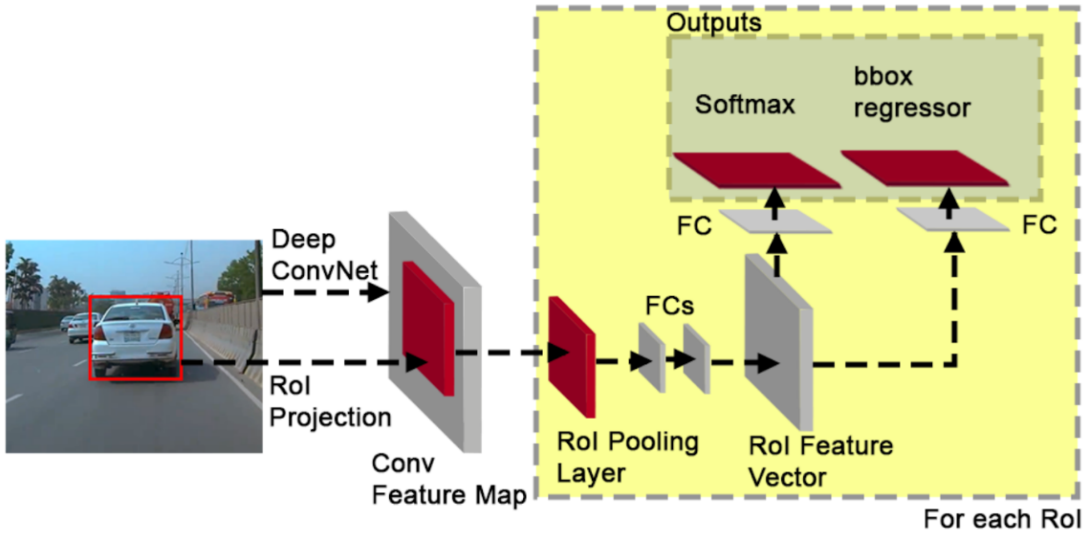


Figure 4.4: R-CNN Working Process

Before sending a picture through a network, as seen in the Figure 4.4, we must extract area suggestions or regions of interest using an algorithm such as selective search. The extracted crops must then be resized, warped and sent across a network.

Finally, a network assigns a category for a specific crop from C+1, including the 'background' label. It also predict which delta Xs and Ys will form a specific crop.

Mixing regions based on pixel intensities, Selective Search is a technique for object localization that combines suggested regions. As a result, pixels are organized hierarchically. The authors extract more than two thousand proposals from the original paper.

Positive vs. Negative examples: Once we have extracted our area suggestion, we must label it for training. As a result, the authors classify all proposals with IOUs of at least 0.5 that have any of the ground-truth bounding boxes. All other area suggestions with an IOU of less than 0.3, on the other hand, are designated as background. As a result, the remainder of them are just ignored. Bounding-box regression [22]:

$$
\begin{aligned}
t_x &= (G_x - P_x) / P_w \\
t_y &= (G_y - P_y) / P_h \\
t_w &= \log (G_w / P_w) \\
t_h &= \log (G_h / P_h)
\end{aligned}
$$

The deltas that CNN will predict are shown in the equations above. Therefore, the width and height are represented by w and h, respectively, whereas the center coordinates are x and y. G and P stand for the and the ground-truth and area proposal bounding box. It is essential to remember that the bounding box loss is computed using only positive data.

4.2 OpenCV :

OpenCV enables us to identify available slots in real-time. The most delicate aspect of this method is that it relies only on image processing. It is very precise in ideal circumstances and requires less resources to operate. Additionally, it is unaffected by a person's movement across space. If anything as large as a vehicle is there, it will recognize it as occupied; otherwise, it will detect available space. Moreover, we ignore the area where detection is unnecessary. This method should be pretty effective if the camera does not move.

Figure 4.5: Raw image of parking lot (top view) [19]

4.2.1 Region of Interest :

Initially, we must mark every one of these parking blocks as our region of interest. It must be done manually since some spaces are not intended for parking. Not all columns are identical, which is another reason for doing it manually. For instance, at Figure 4.6 the second column has a parking spot for a trolley. In addition, the top right corner of the third column lacks parking and is separated by pavement. The advantage of OpenCV over YOLO is the simplicity of installation in a new parking lot. Once set up, we do not need to train anything, and the results are precise. It is an effective technique for realizing that we do not always need a highly complex solution to solve such a problem.

Figure 4.6: Marking region of interest

4.2.2 Thresholding :

Now, to identify available slots, we must do some thresholding. After obtaining each frame from the video, we must convert that frame to grayscale and blur that image for better accuracy. Then, each frame must be converted into a binary image. To do this, we have to use adaptive threshold.

4.2.3 Binary Image Conversion :

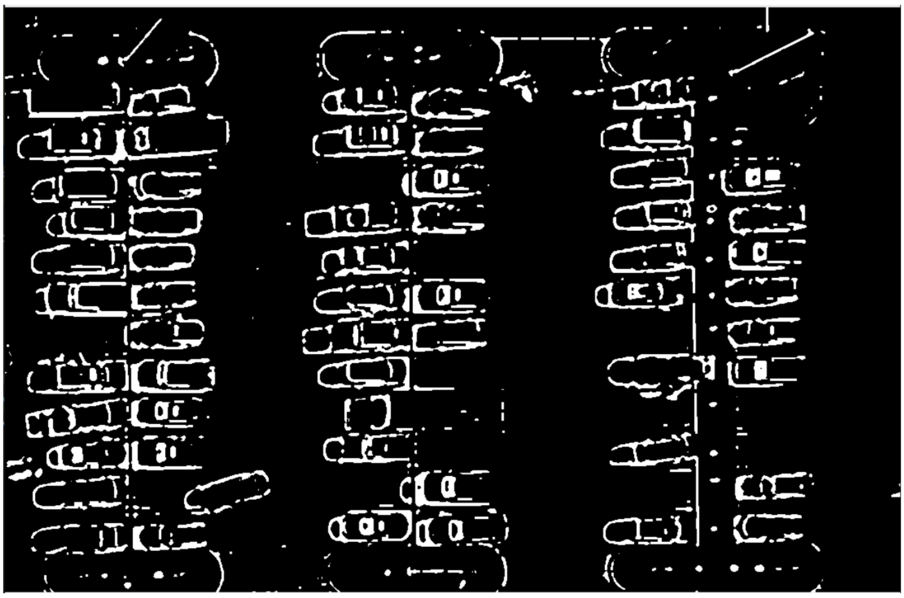


Figure 4.7: Converted frame into binary image

After converting the frame into a binary image as shown in Figure 4.7, if we take a closer look, we can notice that anytime a vehicle is there, a lot of white pixels are visible. On the other hand, when there is no vehicle, there are fewer white pixels. Therefore, this is a strong indicator of the presence or absence of a vehicle.

However, we can see that some white pixel dots remain after the vehicle has disappeared. We can eliminate these pixels using median blur.

```
imgMedian = cv2.medianBlur(imgThreshold, 5)
```

Here, the kernel size is 5. As its size increases, more white pixels will be deleted. We found ”5” as the optimal value for our condition.

4.2.4 Observation :

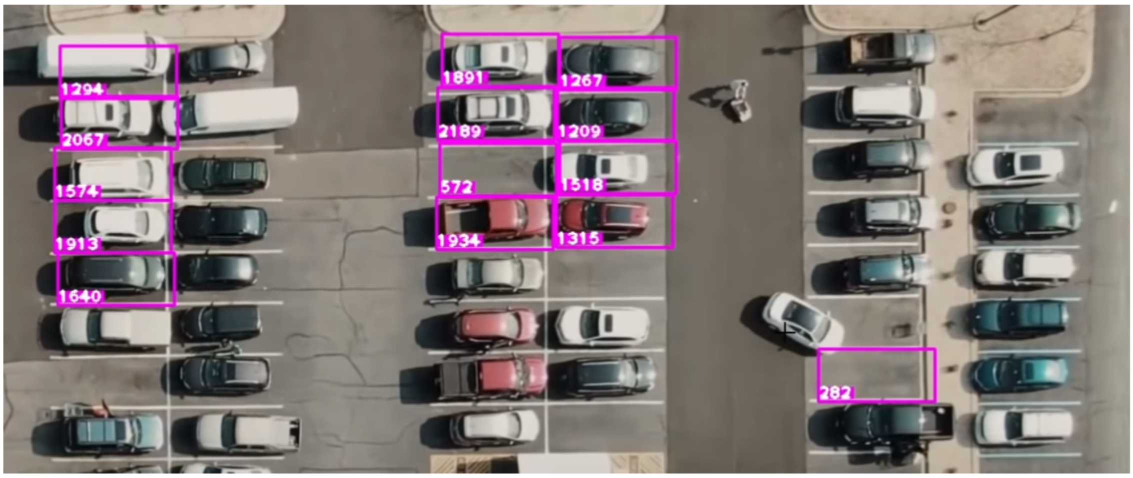


Figure 4.8: Threshold count on marked blocks

Here, we can see in Figure 4.8 when there is a car in the slot, giving us a value of >1000. On the other hand, when there is no car, the threshold value is <600. Nevertheless, to be safe, we can set the threshold for empty slots to <900. So, in this condition, we can say if the count is less than 900, then the slot is empty and if >900, there is a car present in that slot.

4.2.5 Resource Usage :

PID	Resident Me	%Memory	User	%CPU
31655	141.7 MiB	1.8	mint	233

Figure 4.9: CPU and Memory usage of pycharm running OpenCV

Chapter 5

Data Analysis

5.1 Dataset

The dataset comprises car parking lot images that contain occupied and unoccupied parking areas of different parking lots. The dataset consists of different vehicle models and colors in different weather, including sunny days and cloudy days, to train our model. The PKLot and Parking Lot Dataset [6] was examined. The Parking Lot Dataset contains about 18578 photos of parking lots. Similarly, the PKLot dataset contains roughly 12416 photos of parking slots collected from numerous parking lots and categorized on training, validation, and testing sets. Among them, the PKLot data set gave the best result in our experiment.

Moreover, we have used the Plate Recognition Image Dataset [7] of version Origin to train our model to detect vehicle license plates. The dataset contains almost 2658 images which have been further categorized on the train, validation, and test sets.

5.2 Input Data

In Figure 5.1 show different types of vehicles in various parking lots. Among them, some areas are occupied by vehicles, and some are available for parking. The images above depict various weather situations, including bright, overcast, and rainy days, during which the model has been well-trained.

Parking Lot Dataset pictures are usually squares that cannot be turned and are incapable of covering the parking space volume precisely or totally. However, PKLot extracts photos using rotated rectangular masks through layers of algorithm that eventually shroud the parking area pretty precisely. By training and testing both algorithms several times with two completely different datasets, we could compare and evaluate them.

Figure 5.1: Photos taken in different weather and light conditions [6]

5.3 Attributes

The excel sheet consists of the system code number, capacity, occupancy, last update, occupancy rate, date, time, and day of the week.

SystemCodeNumber	Capacity	Occupancy	LastUpdated	OccupancyRate	Date	Time	DayOfWeek
PKLotCam0001	577	61	10/4/16 7:59 AM	10.57192374	10/4/2016	7:59:42	Tue
PKLotCam0001	577	64	10/4/16 8:25 AM	11.09185442	10/4/2016	8:25:42	Tue
PKLotCam0001	577	80	10/4/16 8:59 AM	13.86481802	10/4/2016	8:59:42	Tue
PKLotCam0001	577	107	10/4/16 9:32 AM	18.54419411	10/4/2016	9:32:46	Tue
PKLotCam0001	577	150	10/4/16 9:59 AM	25.9965338	10/4/2016	9:59:48	Tue
PKLotCam0001	577	177	10/4/16 10:26 AM	30.67590988	10/4/2016	10:26:49	Tue
PKLotCam0001	577	219	10/4/16 10:59 AM	37.95493934	10/4/2016	10:59:48	Tue
PKLotCam0001	577	247	10/4/16 11:25 AM	42.80762565	10/4/2016	11:25:47	Tue
PKLotCam0001	577	259	10/4/16 11:59 AM	44.88734835	10/4/2016	11:59:44	Tue
PKLotCam0001	577	266	10/4/16 12:29 PM	46.10051993	10/4/2016	12:29:45	Tue
PKLotCam0001	577	269	10/4/16 1:02 PM	46.62045061	10/4/2016	13:02:48	Tue
PKLotCam0001	577	263	10/4/16 1:29 PM	45.58058925	10/4/2016	13:29:45	Tue
PKLotCam0001	577	238	10/4/16 2:02 PM	41.24783362	10/4/2016	14:02:47	Tue

Figure 5.2: Annotation of number plate

5.4 Data Classification

We split the dataset for our study into 9:1. Which means, 71 percent of the data were used to train our model, 10 percent were used to test it, and the remaining 19 percent data were used for validation.

5.4.1 Training Set

The step in which the machine learning algorithm process is fed labeled example data with answers or output labels. The more data sent into the model throughout the training process, the better the accuracy is. We used roughly 8700 data points from our PKLot dataset to train the model. Also, we have used roughly 1900 images of the Plate Recognition [7] Image Dataset to train our model.

5.4.2 Testing Set

In some cases, the algorithm may learn particular properties of the training set as it iterates to improve performance, using a series of examples for real-world study. Good results for an unknown test collection will increase the likelihood that the algorithm will produce correct answers in the actual world. A total of 1200 pieces of data were used to test our model. Also, we have used roughly 266 images of the Plate Recognition Image Dataset [7] to test our model.

5.5 Data Labels

The dataset can be classified into two labels in this case. There are two types of parking: occupied and available. As a result, we may use a binary classifier to represent our PKLot dataset. Again our Plate Recognation Image Dataset is classified based on City code, metropolitan, Vehicle class, and number.

5.6 Data Visualization

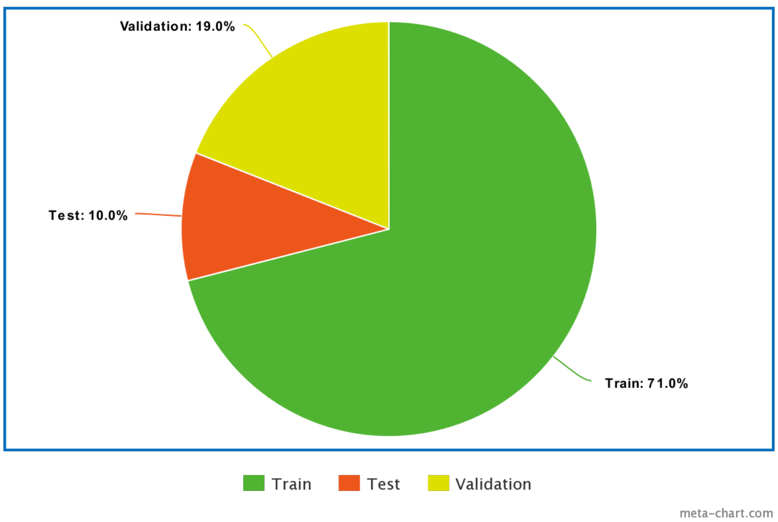


Figure 5.3: PKLot

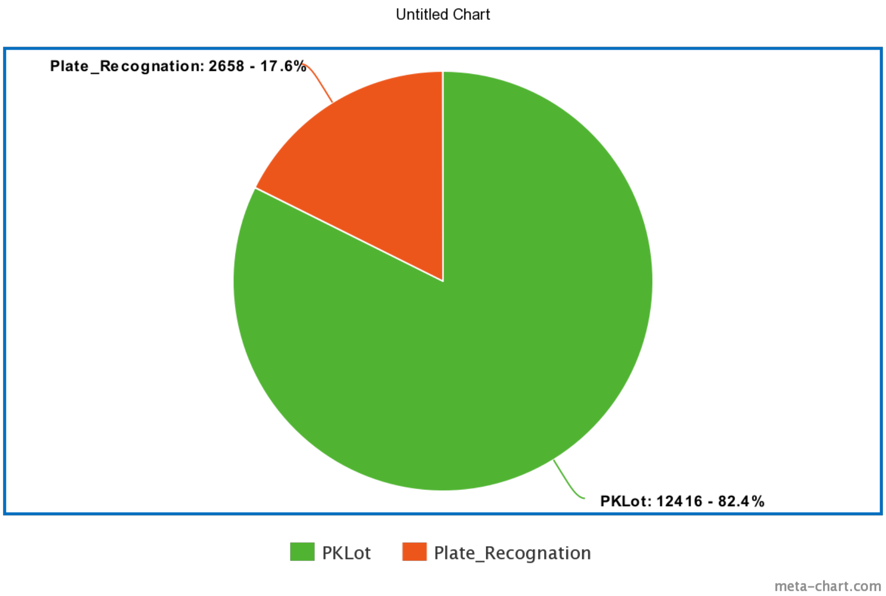


Figure 5.4: Total

Different types of data in the data set are illustrated in the data visualization, and the occupancy rate is shown in Figure 5.3 and 5.4.

5.7 Data Pre-Processing

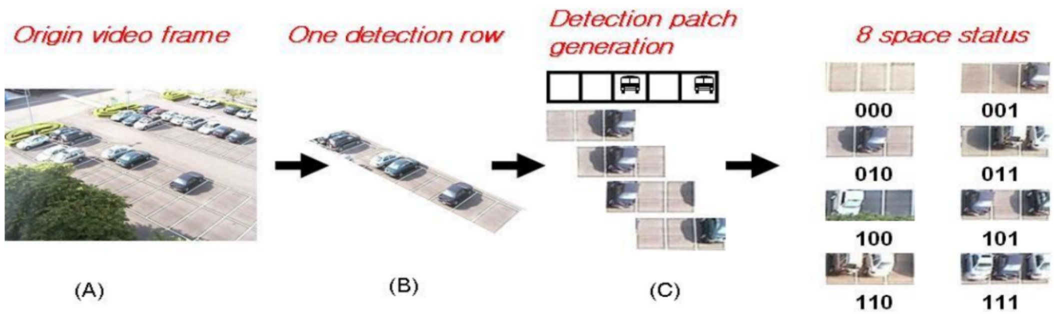


Figure 5.5: Generate detecting patches by preprocessing the input frame

We concentrate on the parking regions when given an input video frame like Figure 5.5. They are simple to acquire because we know the 3D points of an actual scene as well as the intrinsic and extrinsic camera settings. As a detecting patch, one may choose a full row or a single parking space intuitively. However, When a single space is used for detection, the effects of light variation, vehicle shade, and blockade are piled in rows and cannot be reduced. As a result, a detecting patch of three parking spots is used, two of these are shared with neighboring spaces. The original patches have been normalized into rectangular ones after rotation and interpolation. In addition, we manually classify them into 8 space statuses throughout the training phase by labeling empty space with 0 and occupied space with 1.

In addition, during license plate pre-processing, the image is first brightened. The picture is then filtered and divided into individual characters. Following these procedures, the license plate is recovered.

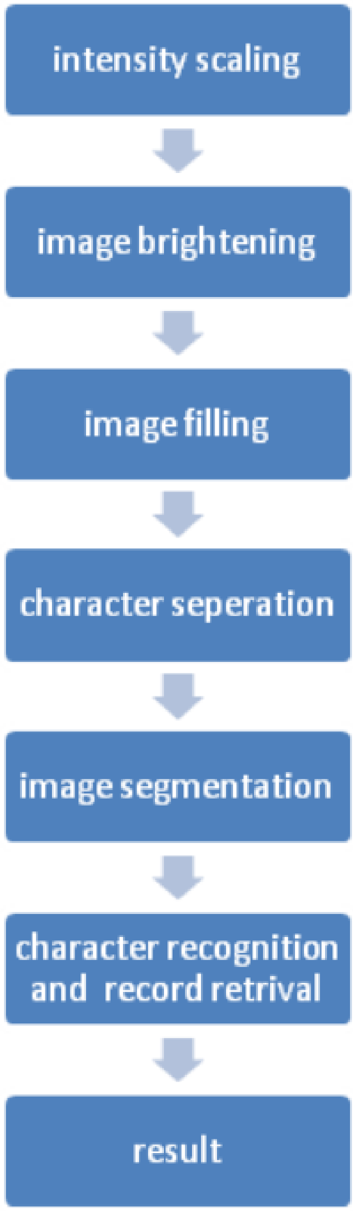


Figure 5.6: License Plate Pre-processing

Chapter 6

Hardware Components

6.1 Raspberry Pi 4

We selected this Raspberry Pi 4 Model B shown in Figure ?? to serve as the system's computer. The current model of the widely used Raspberry Pi is available. It offers vast enhancements in processing speed, graphics, capacity, and connectivity, but it is still compatible with older systems and uses less power. This computing unit can match the performance of entry-level x86 PC systems on a desktop.

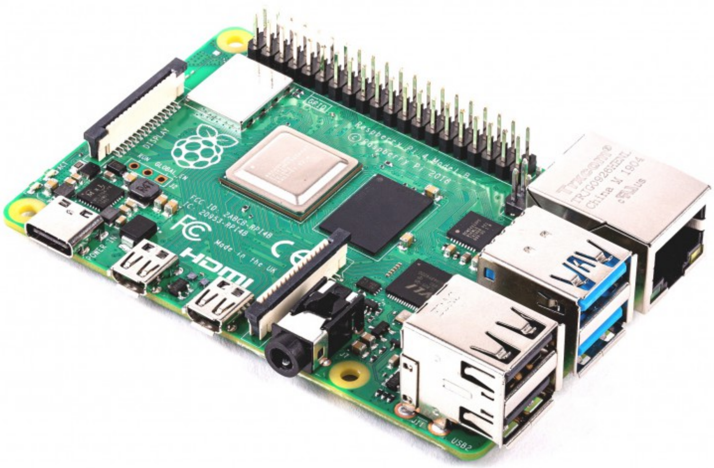


Figure 6.1: Raspberry Pi [23]

One of the most exciting things about this computer is its 64-bit quad-core high-performance Cortex-A72 (ARM) processor. It can also support up to two displays with a maximum resolution of 4K using two micro HDMI ports, and it has a built-in 18 video decoder that works up to 4Kp60. It also has a maximum of 8 GB of RAM and works with USB3.0. It has a wireless LAN with two bands. It also has a license for Bluetooth version 5.0 and a gigabit LAN port already built in.

With dual-band wireless LAN and Bluetooth, the board can be built into finished devices with much less testing for compliance. This saves money and speeds up the process of getting the product on the market.

6.2 High-Quality Camera

Figure 6.2: High-Quality Camera [24] [25]

This camera on Figure 6.2 is the most recent camera add-on that Raspberry Pi has made. The sensor and resolution are both bigger. The new version has a resolution of 12 megapixels instead of 8 megapixels, and its larger sensor, which is about 50% bigger than the old one, makes it better in low light than the old version. It works with C-mount and CS-mount lenses from different manufacturers. Lens converters from a third party make it possible to use lenses of different sizes.

This High-Quality Camera is a solution to the Camera Module v2 for both commercial and industrial applications that require the best image quality and inclusion with highly specialized camera lenses, like surveillance cameras.

A Sony IMX477 sensor PCB board, an FPC connection for a Raspberry Pi computer, a precision-machined lens mount with an inbuilt adjustable mount and a focus correction ring, as well as a C-to-CS adapter, are all included in the box. It includes a mounting component.

6.3 Touch Display

This 7-inch touchscreen display for Raspberry Pi in Figure 6.3 enables the creation of interactive applications which will work as the dashboard for the system.

The touchscreen can take up to 10 multi touch inputs at once, therefore the Raspberry Pi's OS, commonly known as Raspberry Pi OS, provides drivers for the touchscreen. Because of this, our system will not require any external keyboard or mouse.

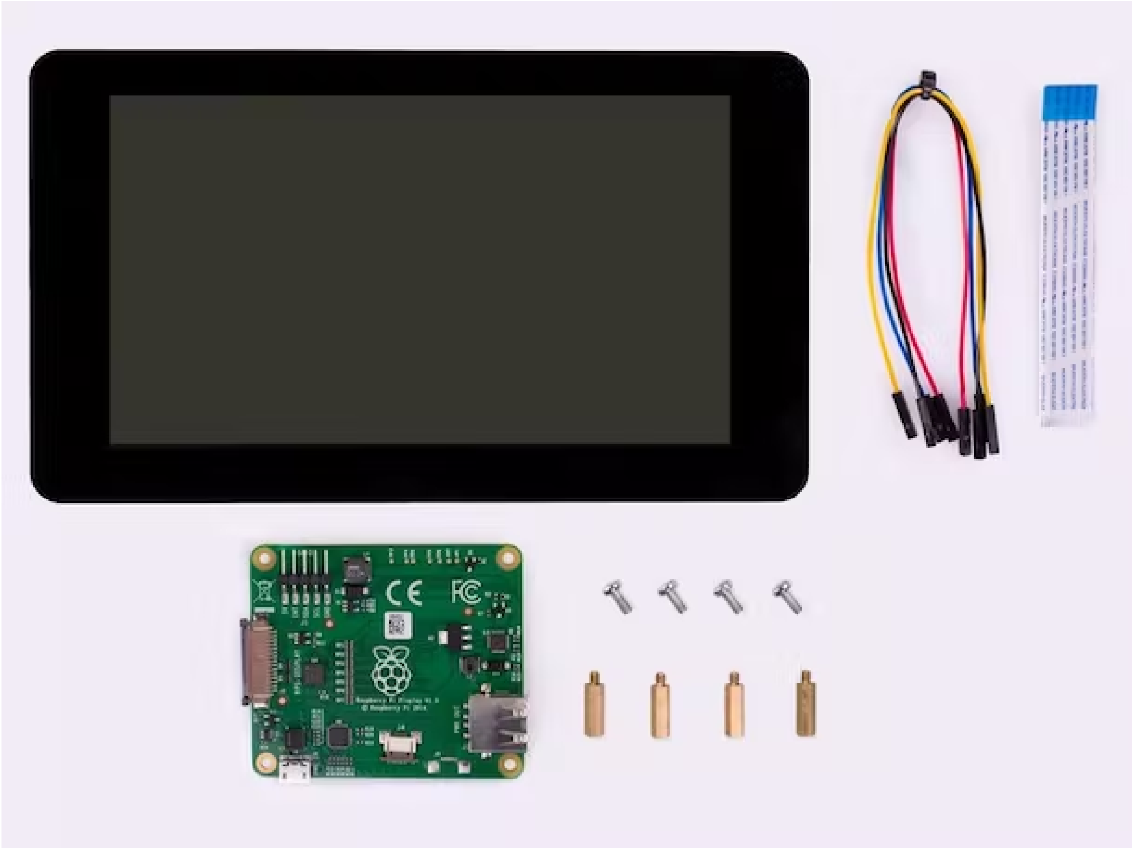


Figure 6.3: 7-inch touchscreen display [26]

The display's resolution is (800*480). It is connected to an adapter board that works as a power supply and handles single transmission between the display and Raspberry Pi. The Raspberry Pi requires just two Connections. The display gets its power from the GPIO port and a combination of multiple wires which connects to the DSI port.

Chapter 7

Platforms, Languages, and Libraries

7.1 Project Requirements :

7.1.1 Functional Requirements :

User:

- Number Plate Detection
- Searching for available slots
- User assigns the booked car to an available slot.
- Token generation
- Money-receipt generation for the user

Admin:

- Admin can see all the available slots at once.
- Admin can look for all the details of the assigned car.

7.1.2 Non-Functional Requirements :

- Responsive design for both mobile and web
- Have proper authentication for admin panel with JWT (sha256 hash encrypted)
- HTTPS (TLS certification)
- Load Balancing access the globe

7.2 UI and UX :

Users can only book the slot, the camera automatically looks for the number plate and available space where it can assign the car change.

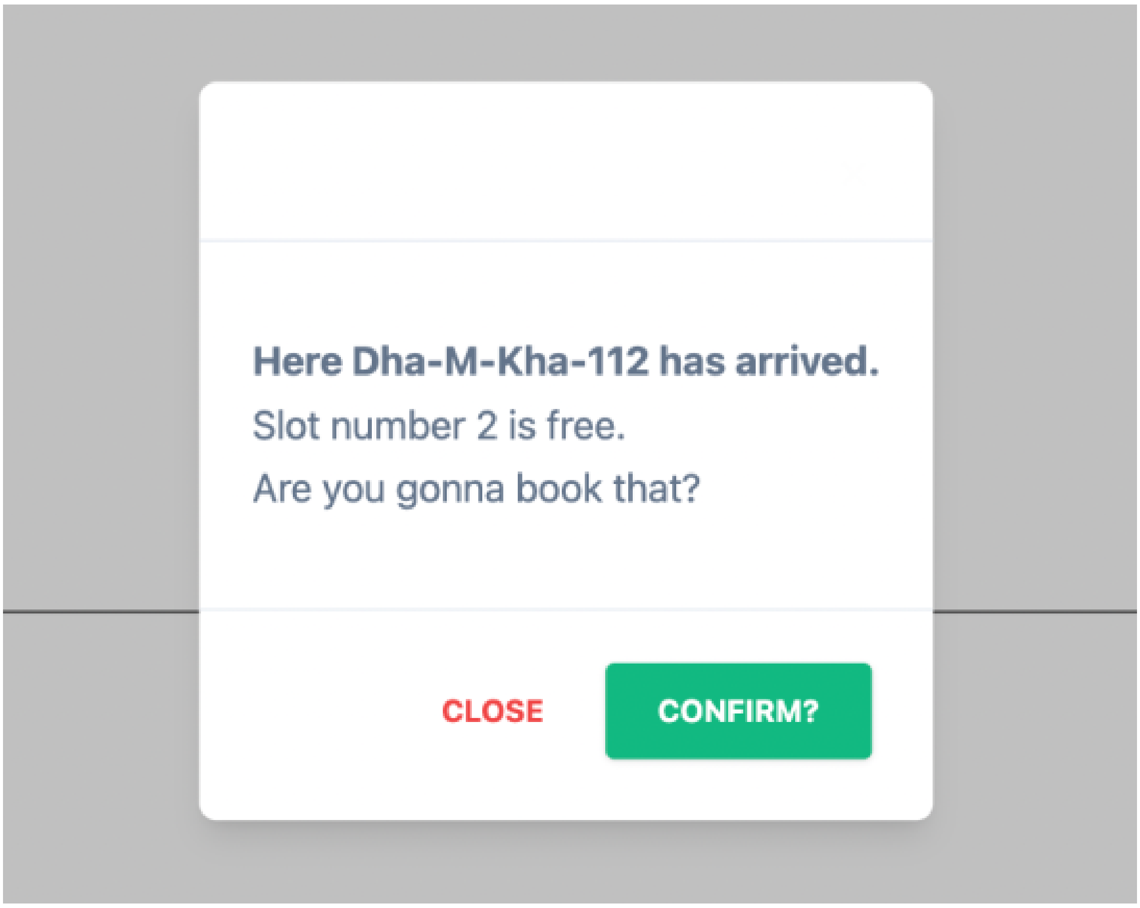


Figure 7.1: Slot Booking Confirmation

The system already knows about the car number plate and the available slot. So if users press the confirm button showing in Figure 7.1, it will confirm the booking of that available slot.

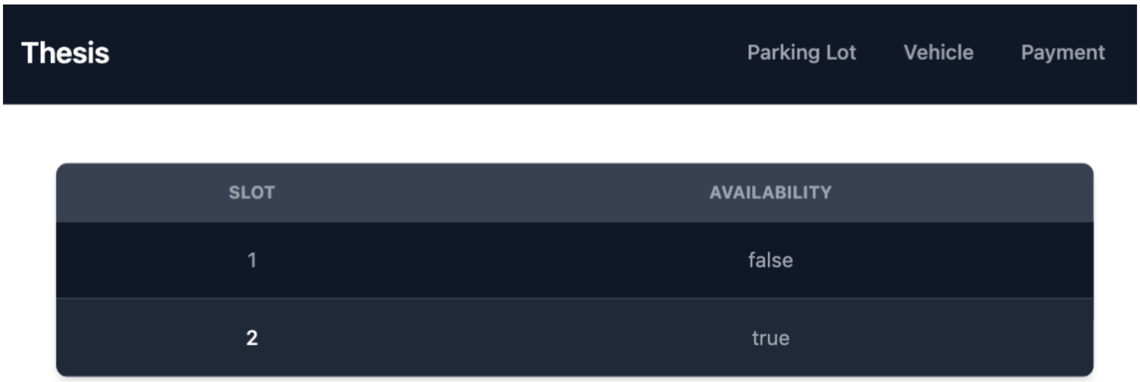
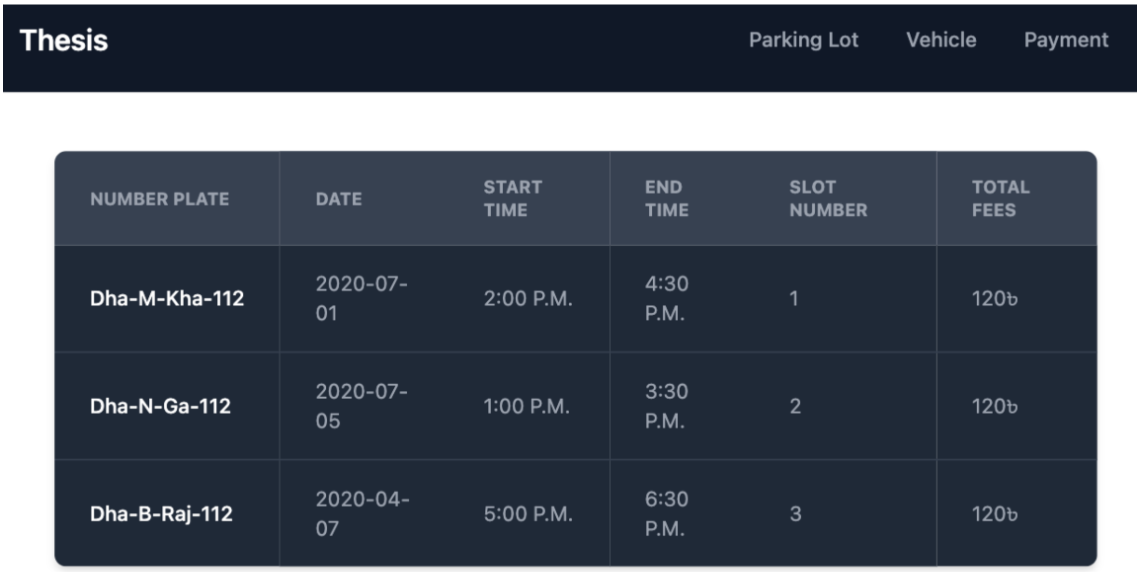


Figure 7.2: Slot Status

In the admin panel in Figure 7.2, it shows the slots and whether they are available or not from a boolean type.



NUMBER PLATE	DATE	START TIME	END TIME	SLOT NUMBER	TOTAL FEES
Dha-M-Kha-112	2020-07-01	2:00 P.M.	4:30 P.M.	1	120৳
Dha-N-Ga-112	2020-07-05	1:00 P.M.	3:30 P.M.	2	120৳
Dha-B-Raj-112	2020-04-07	5:00 P.M.	6:30 P.M.	3	120৳

Figure 7.3: Detail Overview

Furthermore, from the admin perspective, The admin panel will show all the available slots, if they are free or not, and also admin can see the details of the which are already assigned, like when it arrives, when it is the end time, and the slot it has been assigned as shown in Figure 7.3.

7.3 Software Architecture :

We have used MVC pattern [20] to build this project.

View: The whole functionality of the software's user interface is handled by the View component. All Interactive elements have been organized here, such as text, images and dropdown menus. Which the end user will interacts with.

Model: The model element links to the user's overall data-related logic. It could serve in for data associated with business logic or data that is sent back and forth between the View and Controller components. For example, Customer data is pulled from the database via a customer object, changed, and either used to present the data or updated back into the database.

Controller: In order to handle business logic and incoming requests, modify information using the Model component, and connect with Views to produce the result. Controllers serve as a link between both the Model and View components. As an example, the Customer Controller will approach all tasks and inputs from the viewpoint of the customer, and it will utilize the Customer Model to update the database. The client data will be seen by the same controller.

7.4 Programming Languages and Database:

Python: An interpreted, high-level programming language is called Python. The Python Software Foundation helped make Python a reality in 1991. It is mostly used in the creation of websites, software, machine learning, and other related fields.

MongoDB (NoSQL Database): A multi-cloud database solution designed for security, privacy, and scalability. One can create quick, relevant search experiences without a separate search engine. One can give their data life and get real-time insights with embeddable dashboards and visualizations. Utilize a cost-effective, fully managed storage solution to run high-performance analytics. With the command line, create and administer the MongoDB Atlas database. Search, modify, and aggregate data from one or more Atlas databases and AWS S3 buckets without difficulty. Create applications, combine services, and link to data without additional administrative burden. Maintain the accuracy of the data across devices, users, and the back-end.

JavaScript: Another popular language mostly used for web development.

MERN stack is to build both the front-end and back-end. Additionally, it will use PyTorch for machine learning data processing.

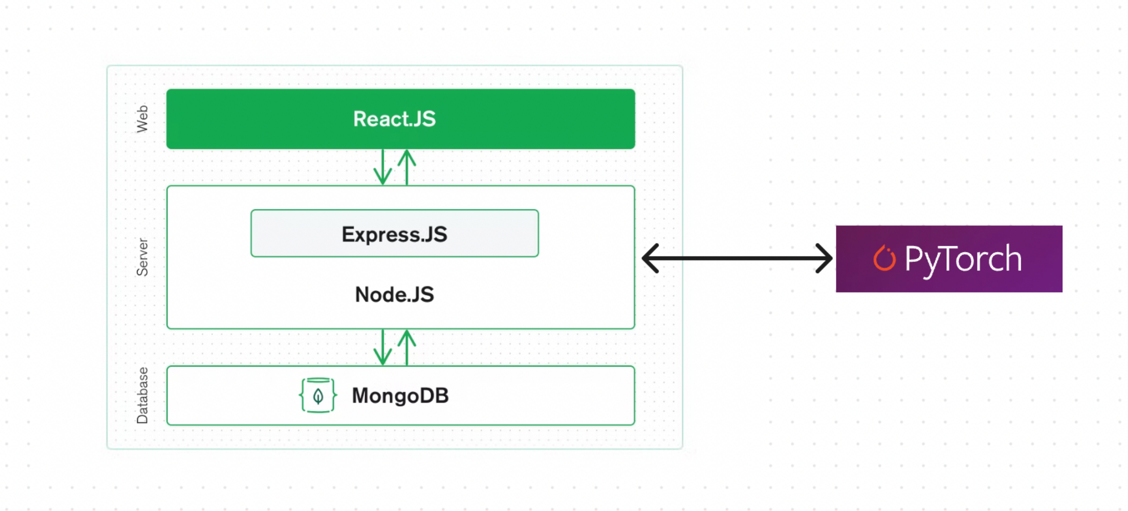


Figure 7.4: Software Development Life Cycle [27]

7.5 Unit Testing :

Software quality assurance is a way of determining if the actual application solution satisfies goals and ensures that it is error-free. It includes running software or system components manually or automatically to examine one or more properties. In contrast to fundamental requirements, discovering defects, gaps, or missing requirements is the responsibility of software testing. Software testing may sometimes be divided into Black Box Testing and White Box Testing [21].

We have tested our application using JEST. It is a framework for testing JavaScript-based web applications.

7.6 Cloud Deployment :

Cloud deployment refers to the implementation of one or more cloud models, such as SaaS, PaaS and IaaS. This involves organizing, administering, and deploying cloud-based workloads.

- Deployments that really are easier and more efficient. Automatic data service procurement along with releases that distribute updated versions of the software, database systems, and applications.
- Sell price reductions. Utilize consumption-based pricing to reduce expenses and get rid of on-premises setups with high capital expenditures.
- Implementation processes have used the across the globe architecture offered by CSPs to extend company's reach into the other regions of the world.
- Modern digital business architectures. Utilize the features and capabilities that CSPs are continuously delivering, promote new technologies, and develop creative digital business models.
- Adaptability in business. Applications should be built for the high availability that Security mechanisms provide in order to make the enterprise robust.
- Agility and scalability Using autoscaling and scalability to handle peak demand without reserving additional capacity.
- Geographical scope Use any device or location to access apps using the CSPs' robust connection infrastructure
- Operative effectiveness. Utilize the cloud's inherent automation to boost productivity and minimize labor-intensive tasks.

Considering AWS's cloud platform is the most sophisticated and well enough in the globe, AWS S3 is the wisest option for deployment. Over 200 properly operational programs are accessible by AWS from server farms located throughout the world.

S3 could be used for a multitude of front-end systems to provide consumers access to the source code data. The front-end server was identified by a DNS Network infrastructure that uses load balancing. The front-end servers' load dictates which IP address to use when a client asks the IP address of either an S3 bucket.

Chapter 8

Methodology

8.1 Working Process

The purpose of smart parking using ML and the cloud is to optimize parking. The model requires designing a process that takes data from the user, and the system will process data systematically and produce the result.

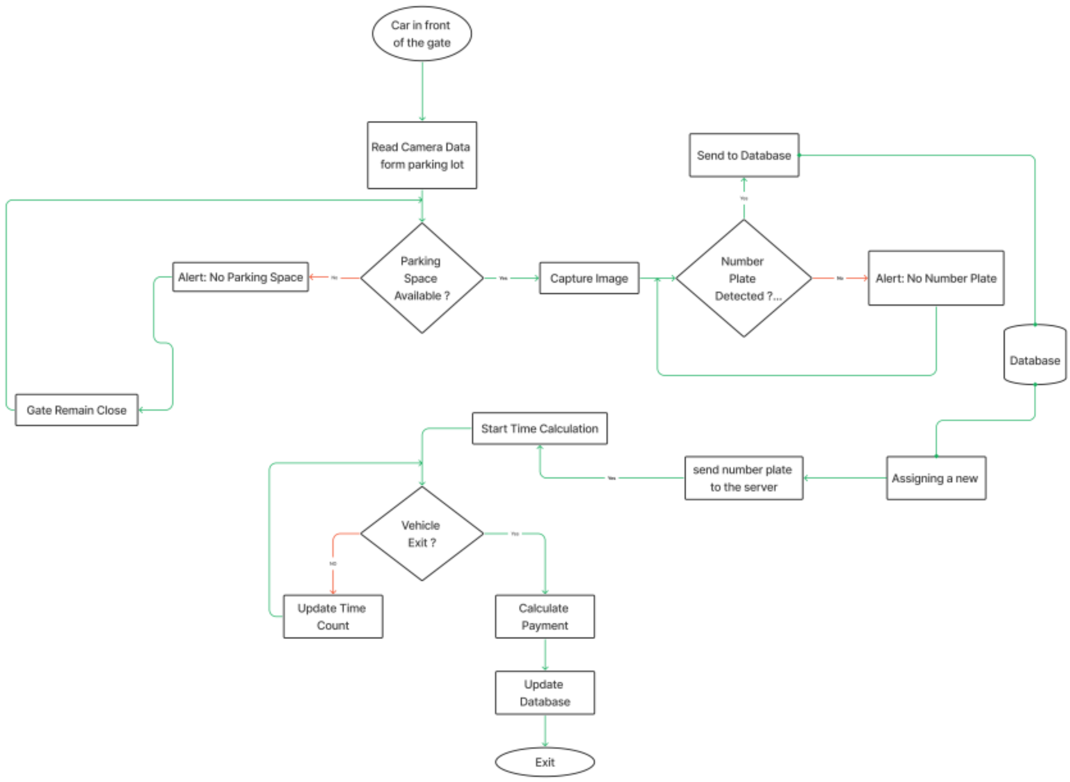


Figure 8.1: Flow-chart of the proposed intelligent parking model

A Smart Parking system is a process that is the result of the increasing amount of cars. The Process Requires Data Collection and Data Modification, the process consists of six main processes.

- It takes Vehicle Information such as Detects license plates and stores them in a database.
- Then, the data is processed to give Suggestions to the user.
- It uses Cameras To take Images and Send Data to a Cloud-Based Database to process the data with Machine Learning algorithms.
- Optimal Resource Allocation and Reservations are done through Cloud Computing and Machine Learning.
- Calculate the time of using any slot based on Base fair + (exit time - entry time).
- Conducting Payment Process.

When a vehicle arrives in a parking lot, the display will show the user whether any parking lot is available or not. If no space is available, the system won't let the vehicle enter. If space is available, in front of the gate there will be a stop position for a moment to complete the parking process. During this moment, the system will take a picture of the driving license plate; then it will process the image and send it to the database. After that, the system will appoint a slot for the car and give information about the position to the driver.

Meanwhile the timer will be started for that vehicle. When the driver is leaving, the exit gate will also stop for a moment. During this time, the system will again take a picture of the number plate, and the system will calculate the total time and cost. The total cost will be shown on the screen. The user can pay the payment in both ways: cash or cashless(via app or card). After payment, the vehicle will leave the parking lot, and the system will update its counter for that parking space.

Chapter 9

Result

Following the training of a large dataset on two distinct methods to determine the efficiency of the algorithm in detecting the existence of an item (Vehicle) in the input data (.jpg,.mp4,.csv), the following four keywords are used:

True Positive(TP) → unfilled spaces are accurately categorized
False Positive(FP) → filled spaces were wrongly categorized
True Negative(TN) → non-empty spaces were accurately categorized
False Negative(FN) → non-empty spaces were mistakenly categorized

Where, The letter P denotes total positive numbers. And the total negative numbers are denoted by the letter N.

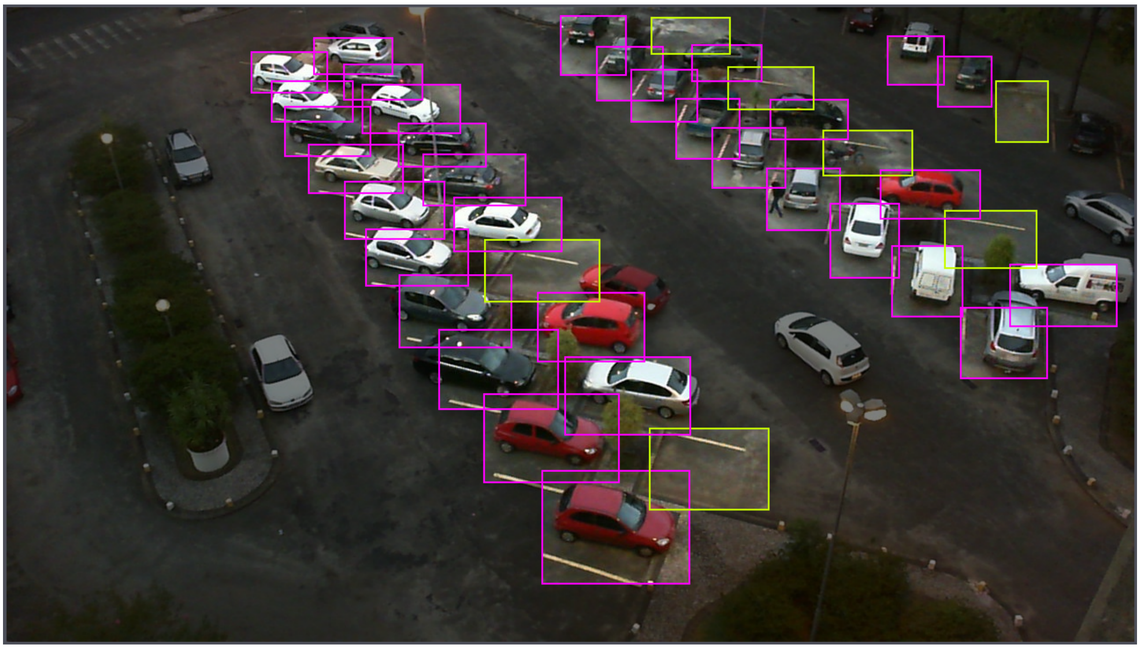


Figure 9.1: Purple Boxes Are Occupied And Yellow Boxes Are Empty [6]

The findings of this study were achieved using two algorithms, R-CNN and YOLOv5, to classify parking spots as free or occupied. We used the PKLot Dataset to test our technique. Precision Score, Recall Score, and F1 Score are the three criteria used to evaluate performance, efficiency, and effectiveness. The algorithm's efficiency increases as the value of these parameters rise. Images of both free and occupied parking spaces are included in this dataset. The dataset is split in portions for training and testing, with 71 percent of photos utilized for training; 19 percent for validation and 10 percent for testing. We were able to collect very exact results and show them in the form of tables and graphs.

PKLot		
MATRICES	YOLO-V5	RCNN
TP	8365	8213
FP	335	487
TN	8121	8053
FN	98	178

Figure 9.2: Matrices For Pklot Dataset

Precision Score : It's the proportion of correctly categorized blank spaces to the total number of successfully identified and mistakenly identified blank spaces.

precision = TP/(TP+FP)

ALGORITHM	DATASETS
	PKLot
YOLO-V5	0.961
RCNN	0.944

Figure 9.3: Results For Precision Score Parameter

Recall Score : Using the Recall score formula, which is the ratio of properly recognized blank spaces to the total of correctly identified empty spaces and mistakenly identified non-empty spaces.

Recall = (TP/(TP+FN))

ALGORITHM	DATASETS
	PKLot
YOLO-V5	0.988
RCNN	0.978

Figure 9.4: Results For Recall Score Parameter

F1- Measure Score : It's the proportion of correctly categorized empty spaces to the total number of correctly classified empty spaces, divided by half of the total number of mistakenly recognized empty spaces and non-empty spaces.

F1 Measure = (TP/(TP+0.5(FP+FN)))

ALGORITHM	DATASETS
	PKLot
YOLO-V5	0.974
RCNN	0.961

Figure 9.5: Results For F1 Score Parameter

Precision, recall, and F1 scores were compared across datasets in the Table 9.6 below.

Method	Accuracy	Precision	Recall	F1 Score
RCNN	0.960	0.944	0.978	0.961
YOLO-V5	0.974	0.961	0.988	0.974

Figure 9.6: YOLOv5 vs R-CNN Comparison

YOLOv5 is determined to be superior to R-CNN in each of the three scenarios as shown in Table 9.6. Though the figures appear to be similar, YOLOv5 takes the lead when assessing the best of the options.

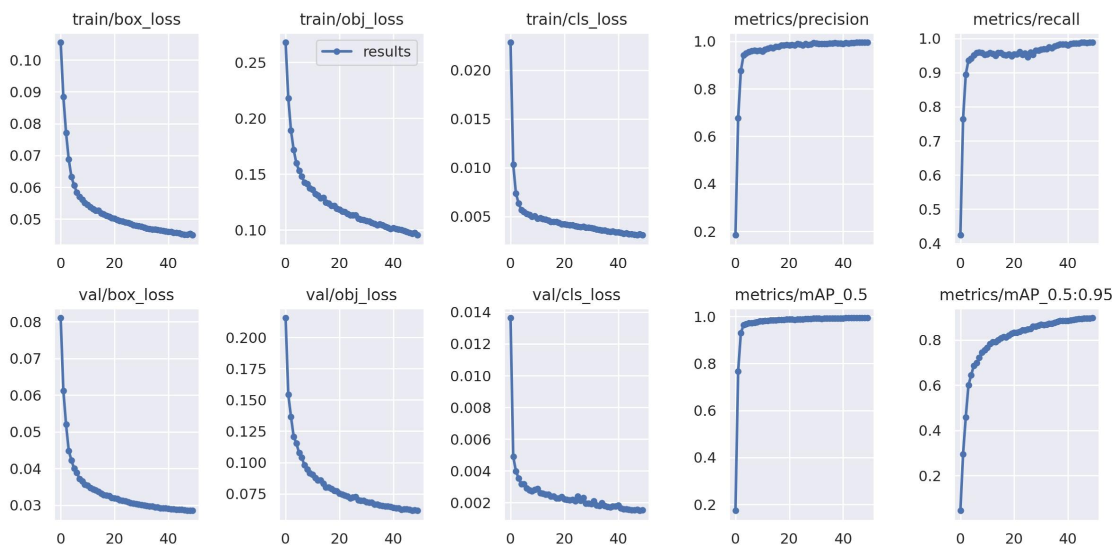


Figure 9.7: Model Training result for parking lot detection (YOLOv5)

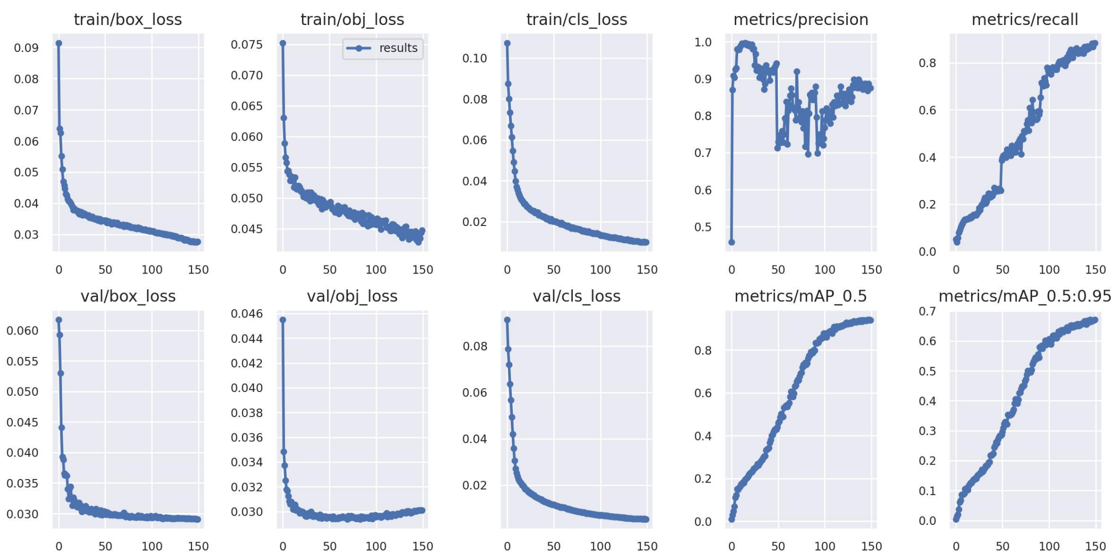


Figure 9.8: Model Training result for Number Plate Recognition (YOLOv5)

9.1 OpenCV :

Here in in Figure 9.9 the green rectangles indicate available slots and red represents occupied.

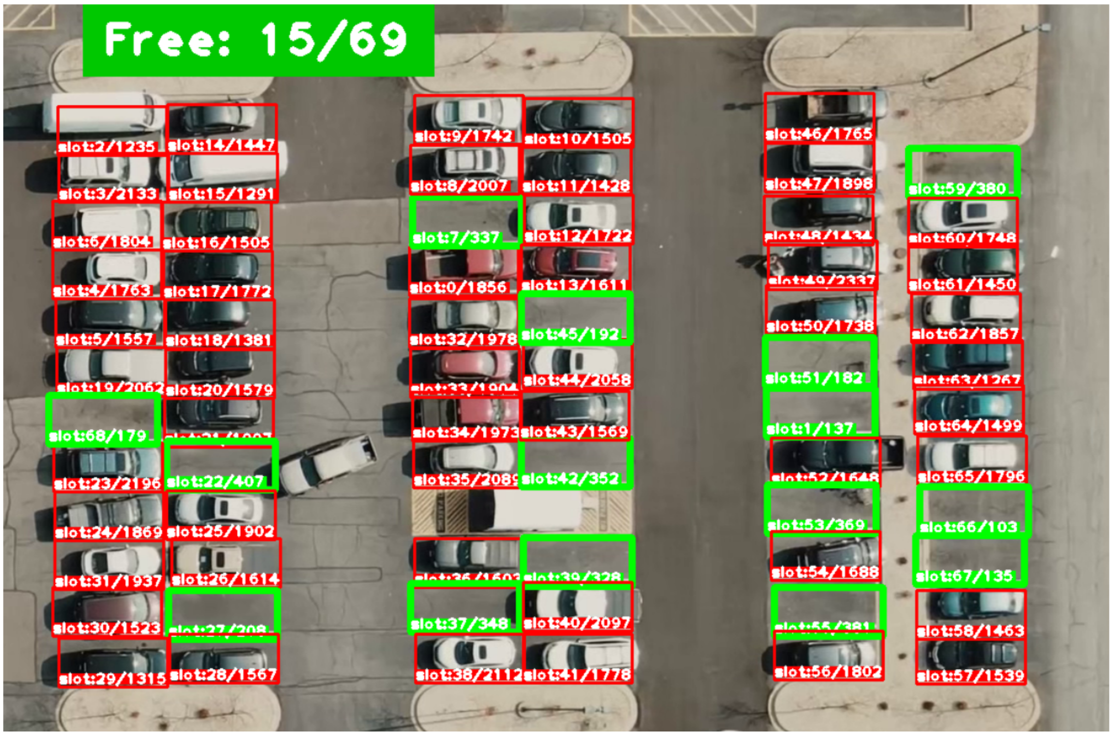


Figure 9.9: Updating slot status in real time

We found out that OpenCV uses less CPU and Memory than YOLOv5. It can run flawlessly on a home PC with 8GB of RAM and without any dedicated GPU.

However, this method cannot be used outside. In the event of a change in lighting, its accuracy declines. Therefore, OpenCV is not appropriate for this purpose.

9.2 Number Plate Recognition :

Detection of vehicle license plates is a critical part of this project. A vehicle's number plate is a unique identity. Based on the license plates of the cars, we may assign parking places and generate charges. In addition, we may use this information in the future to predict the parking lot's occupancy.

Every vehicle in Bangladesh must be attached BRTA standard license plate, both front and rear.

9.2.1 Character Segmentation :

The YOLOv5 model was trained using 1859 images of Bangla Number Plates and 105 classes. Taking the BRTA Standard License Plate configuration into account, each class has been divided.

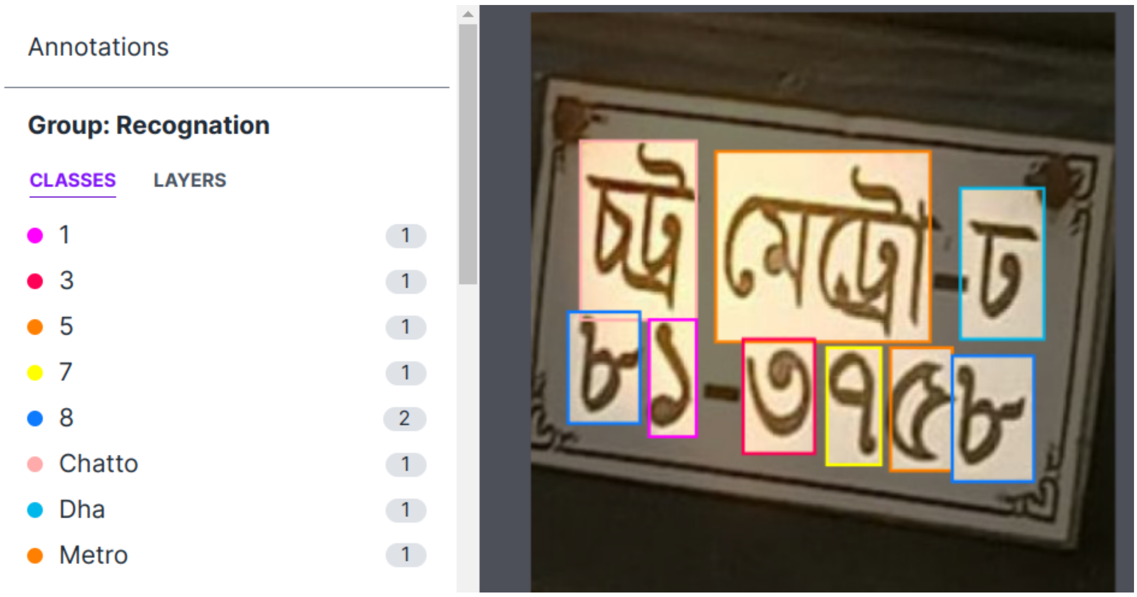


Figure 9.10: Annotation of number plate

In Figure 9.10, Chatto is the city code. Metro is for metropolitan. And Dha is the vehicle class. Lastly, each digit of the number plate has been segmented separately as well.

9.2.2 Model Training :

We have used a publicly available dataset of Bangla license plates. Then, YOLOv5 is used to train this dataset. This collection comprises 2658 photographs of bengali license plates. Our model was evaluated using the Nvidia Tesla T4 GPU at the Google Collaboratory. 69.6 minutes were required to accomplish 149 epochs. With mean Average Precision (mAP50) of 0.951

Figure 9.11: Text Recognition From Number Plate

9.2.3 Model Summary :

157 layers, 7293310 parameters, 0 gradients, 16.7 GFLOPs

9.2.4 Future Advancements :

The dataset used is insufficient to interpret license plate information with 100 % precision. Therefore, more data is required to train this model.

9.3 Conclusion

In today's world, finding a parking system that is simple to use is extremely difficult. Since the introduction of industrialized cities, the number of cars has increased, and individuals are increasingly having difficulty maneuvering their vehicles into parking lots. This parking crisis gives birth to innovative technologies using Machine learning and cloud computing to manage automobile parking systems. Our article proposes an ML-based application solution to alleviate the automobile parking dilemma in modern and remote cities. The suggested solution delivers real-time information about a car parking lot, dynamically assigns slots, and generates payment systems. This research minimizes human resources, maximizes parking system automation, and makes life easier for vehicle users. Thus this research represents an attempt to eradicate the parking problem and reduce the complexity of our daily life with the help of machine learning.In today's world, finding a parking system that is simple to use is extremely difficult. Since the introduction of industrialized cities, the number of cars has increased, and individuals are increasingly having difficulty maneuvering their vehicles into parking lots. This parking crisis gives birth to innovative technologies using Machine learning and cloud computing to manage automobile parking systems. Our article proposes an ML-based application solution to alleviate the automobile parking dilemma in modern and remote cities. The suggested solution delivers real-time information about a car parking lot, dynamically assigns slots, and generates payment systems. This research minimizes human resources, maximizes parking system automation, and makes life easier for vehicle users. Thus this research represents an attempt to eradicate the parking problem and reduce the complexity of our daily life with the help of machine learning.

Bibliography

[1] Rahman, M. S., Mostakim, M., Nasrin, M. S., and Alom, M. Z. (2019, December). Bangla license plate recognition using convolutional neural networks (CNN). In 2019 22nd International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[2] Gautam, S. (2022, March 15). How Illegal Parking on Roads Causes Traffic Jams. Get My Parking. Retrieved September 24, 2022, from https://blog.getmyparking.com/2018/11/13/how-illegal-parking-on-roads-causes-traffic-jams/

[3] Tang, V. W., Zheng, Y., and Cao, J. (2006, August). An intelligent car park management system based on wireless sensor networks. In 2006 First International Symposium on Pervasive Computing and Applications (pp. 65-70). IEEE.

[4] Sabnam, M., Das, M., and Kashyap, P. A. (2016). Automatic Car Parking System. ADBU Journal of Engineering Technology, 4.

[5] Gu, J., Zhang, Z., Yu, F., and Liu, Q. (2012, July). Design and implementation of a street parking system using wireless sensor networks. In IEEE 10th International Conference on Industrial Informatics (pp. 1212-1217). IEEE.

[6] Parking Lot Dataset. (2018, December 5). Kaggle. Retrieved September 24, 2022, from https://www.kaggle.com/datasets/blanderbuss/parking-lot-dataset

[7] Plate_Recognation Object Detection Dataset by new-workspace-fwhq9. (n.d.). Roboflow. Retrieved September 25, 2022, from https://universe.roboflow.com/new-workspace-fwhq9/plate_recognation

[8] Foote, K. D. (2022, January 20). A Brief History of Machine Learning. DATAVERSITY. Retrieved September 26, 2022, from https://www.dataversity.net/a-brief-history-of-machine-learning/

[9] Ancion, K. (2019, March 25). What are the benefits of smart parking for drivers? Parkeagle. Retrieved September 24, 2022, from https://www.parkeagle.com/2019/03/19/what-are-the-benefits-of-smart-parking-for-drivers/

[10] What is a smart parking system? Functionalities and benefits. (2021b, August 17). Retrieved September 25, 2022, from https://tomorrow.city/a/smart-parking

[11] Zhang Bin, Jiang Dalin, Wang Fang, and Wan Tingting. A design of parking space detector based on video image. ICEMI 2009 - Proceedings of 9th

International Conference on Electronic Measurement and Instruments, pages 2253–2256.

[12] BRTA (2020). Retrieved from: https://brta.portal.gov.bd/sites/default/files/files/brta.portal.gov.bd/page/6d849ccb_09aa_4fbe_aef2_3d254a2a0cd1/2020-07-02-23-21-fba1ebaa3c6a7299fed0d5c2ab8f32fa.pdf

[13] Morillo Carbonell, C., and Campos Cacheda, J. M. (2016, June 7). EFFECT OF ILLEGAL ON-STREET PARKING ON TRAVEL TIMES IN URBAN ENVIRONMENT. Libro De Actas CIT2016. XII Congreso De Ingeniería Del Transporte. https://doi.org/10.4995/cit2016.2016.3521

[14] Who invented the automobile? (n.d.). The Library of Congress. Retrieved September 24, 2022, from https://www.loc.gov/everyday-mysteries/motor-vehicles-aeronautics-astronautics/item/who-invented-the-automobile/

[15] Fernando Enr´ıquez1, Luis Miguel Soria1, Juan Antonio A´ lvarez-Garc´ıa1, Francisco Velasco2, and Oscar D´eniz3 (2017). Existing approaches to smart parking: An overview. In Computer Languages and Systems Department, University of Seville, 41012 Seville, Spain. Retrieved from: https://www.researchgate.net/publication/318100909_Existing_Approaches_to_Smart_Parking_An_Overview

[16] Patro, S. P., Patel, P., Senapaty, M. K., Padhy, N., and Sah, R. D. (2020, March). IoT based smart parking system: a proposed algorithm and model. In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA) (pp. 1-6). IEEE.

[17] Widyasari, R., Candra, M. Z. C., and Akbar, S. (2019, November). Iot-based smart parking system development. In 2019 International Conference on Data and Software Engineering (ICoDSE) (pp. 1-6). IEEE.

[18] Quader, S. M., Nova, S. H., Talukdar, S. D., Sayeed, M., and Sadab, M. R. (2021). IoT based parking system: prospects, challenges, and beyond (Doctoral dissertation, Brac University).

[19] Computer Vision Zone. (2022, May 6). Parking Space Counter. Retrieved September 26, 2022, from https://www.computervision.zone/courses/parking-space-counter/

[20] Model-View-Controller Pattern. (2009). Learn Objective-C for Java Developers, 353–402. https://doi.org/10.1007/978-1-4302-2370-2_20

[21] Verma, A., Khatana, A., and Chaudhary, S. (2017). A comparative study of black box testing and white box testing. Int. J. Comput. Sci. Eng, 5(12), 301-304.

[22] Wu, B., Shen, Y., Guo, S., Chen, J., Sun, L., Li, H., and Ao, Y. (2022). High Quality Object Detection for Multiresolution Remote Sensing Imagery Using Cascaded Multi-Stage Detectors. Remote Sensing, 14(9), 2091.

[23] Ltd, R. P. (n.d.). Buy a 4 Model B –. Raspberry Pi. Retrieved September 26, 2022, from https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[24] Ltd, R. P. (n.d.-b). Buy a High Quality Camera –. Raspberry Pi. Retrieved September 26, 2022, from https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/

[25] Foundation, R. P. (2020). Getting started with the Camera Module. Raspberry Pi. Retrieved September 26, 2022, from https://projects.raspberrypi.org/en/projects/getting-started-with-picamera

[26] Ltd, R. P. (n.d.-c). Buy a Touch Display –. Raspberry Pi. Retrieved September 26, 2022, from https://www.raspberrypi.com/products/raspberry-pi-touch-display/

[27] MongoDB. (n.d.). What Is The MERN Stack? Introduction & Examples. Retrieved September 27, 2022, from https://www.mongodb.com/mern-stack