A Project Report on

IoT and ML based Cross Platform Application for Designing Smart Parking

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

Bachelor of Engineering

in

Information Technology

by

Snehal Shanbhag(19104008)
Pranjali Shimpi(19104017)
Akansha Rawat(19104007)

Under the Guidance of

Prof. Sonal Jain
Prof. Charul Singh

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2022-2023

Approval Sheet

This Project Report entitled ***"IoT and ML based Cross Platform Application for Designing Smart Parking"*** Submitted by ***"Snehal Shanbhag"(19104008), "Pranjali Shimpi"(19104017),"Akansha Rawat"(19104007)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Charul Singh Co-Guide	Prof. Sonal Jain Guide

Dr. Kiran Deshpande
Head of Department of Information Technology

Place: A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"IoT and ML based Cross Platform Application for Designing Smart Parking"*** submitted by ***"Snehal Shanbhag" (19104008), "Pranjali Shimpi" (19104017), "Akansha Rawat" (19104017)*** for the partial fulfillment of the requirement for the award of a degree ***Bachelor of Engineering*** in ***Information Technology***, to the University of Mumbai, is a bonafide work carried out during the academic year 2022-2023.

Prof. Charul Singh Co-Guide	Prof. Sonal Jain Guide
Dr. Kiran Deshpande Head of Department of Information Technology	Dr. Uttam D.Kolekar Principal

External Examiner(s)

1.

2.

Place: A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **IoT and ML based Cross Platform Application for Designing Smart Parking.** We take this opportunity to express our sincere thanks towards our Guide **Prof. Sonal Jain** & Co-Guide **Prof. Charul Singh** Department of Information Technology, A. P. Shah Institute of Technology Thane for providing the technical guidelines and suggestions regarding the line of work. We would like to express our gratitude for their constant encouragement, support, and guidance through the development of project.

We thank **Dr. Kiran B. Deshpande** Head of Department of Information Technology, A. P. Shah Institute of Technology for his encouragement during the progress meeting and for providing guidelines to write this report.

We thank **Prof. Sonal Jain** BE project co-ordinator, Department of Information Technology, A. P. Shah Institute of Technology for being encouraging throughout the course and for guidance.

We also thank the entire staff of A. P. Shah Institute of Technology for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of A. P. Shah Institute of Technology for their encouragement.

Snehal Yogesh Shanbhag
(19104008)

Pranjali Sanjay Shimpi
(19104017)

Akansha Manoj Singh Rawat
(19104007)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Snehal Yogesh Shanbhag(19104008)

Pranjali Sanjay Shimpi(19104017)

Akansha Manoj Singh Rawat(19104007)

Date:

Abstract

The term parking management system typically refers to the customized hardware-intensive systems installed in buildings and malls. The project aims to build a cross-platform smart parking system that can solve parking problems by reducing the time for drivers by searching for vacant positions in car parking lots and also providing efficient parking space utilization. Various reports have shown that the smart parking system reduces traffic problems and increases business growth and economic development within that locality. However, there are many places where such expensive solutions cannot be installed due to various reasons, like cost and setup requirements. This project focuses on developing a parking management system based on object detection to detect vacant parking slots in our college premises where automated systems are not installed. Camera images of the parking area are put through an object detection algorithm that detects virtual slots in the area and extracts residency information to guide the potential drivers about the availability and position of empty spaces. A smart parking system that involves a camera connected to an application using cloud computing makes use of a less complex Machine Learning model to detect empty positions. We train and cross-validate our model using the real-time customized dataset and YOLOv5 algorithm. In order to display the reserved spaces based on the data fetched from the model IoT will be used. The project for the backend will be using Cloud Services.

Contents

List of Figures

List of Abbreviations

YOLO:	You Only Look Once
IOT:	Internet Of Things
CNN:	Convolution Neural Network
RNN:	Recurrent Neural Network
SVM:	Support Vector Machine
OLED:	Organic Light-Emitting Diode

Chapter 1

Introduction

With the continuous development of the economy, personal vehicles have become a crucial part of our daily lives. The artifact provides a comfortable way of life and has become affordable to the most working class which has led to an exponential increase in the number of vehicles in the past few years causing multiple problems. The organizations like colleges, universities, and MNCs witness significant influx and outflow of vehicles throughout the day. Vehicle drivers find it difficult to get real-time information about an accessible parking slot and manage the parking of the vehicle because the size of parking slots is getting reduced and the number of vehicles is increasing.

The smart parking occupancy detection system is a technology that aims to mitigate traffic congestion problems by reducing the time for drivers to look for vacant positions in automotive parking lots and providing efficient parking space utilization. Several smart parking solutions employ multiple low-cost cameras to detect vacant parking spots within the area. To detect available parking spots from visual information captured by cameras, many papers proposed using machine learning techniques such as support vector machine (SVM), Bayesian and convolutional neural networks (CNN); however, to date, few have presented acceptable prediction accuracy over a range of visual information.

Various techniques for object detection already exist, they can be split up into two categories, the first category is the algorithms based on Classifications. CNN and RNN come under this category. In this, the model has to select the interesting regions from the image and have to classify them using the Convolutional Neural Network. This technique is very slow because in order to run a prediction for each selected region. The second category of the algorithms is based on Regressions. The YOLO algorithm is another example of this category. The interesting areas of the image won't be chosen in this. Instead, the model predicts the classes and bounding boxes of the complete image at a single run of the algorithm and detects multiple objects using a single neural network. The YOLO creates a grid system out of an image, and each grid finds items within that specific grid. Thus, they are best used for real-time real-world object detection based on the data provided.

The YOLO algorithm is fast as compared to other classification algorithms. In real-time our algorithm processes 45 frames per second. YOLO algorithm makes localization errors but predicts fewer false positives in the background. These algorithms are not tested with degraded images, i.e. they are trained with academic data sets, including ImageNet, COCO, and VOC, etc. but they are not well tested with randomly captured data sets. The main issues of images captured in the real scene are :

- Due to the instability of the camera, the captured images may be blurred.
- The images can also not be clear enough because the object can be obstructed.
- The images may have poor quality as a result of bad weather, overexposure or low resolution.

1.1 Objectives

The primary objective of the smart parking occupancy detection system is to address parking issues by minimizing the amount of time drivers spend searching for available parking spots and optimizing the utilization of parking spaces.

Based on the problems identified, below are our project objectives :

- To identify and locate one or more vacant parking slots for vehicles detected through camera.
- To enable drivers to locate any available parking spots and reserve a particular spot through the App GUI.
- To train and test data using YOLO algorithm for maintaining high accuracy.
- To display the count of vacant and reserved parking on OLED dashboard.
- To monitor parking availability based on high precision and detection of parking space using machine learning and image processing.

Chapter 2

Literature Review

This section aims to understand IoT and ML cross-platform applications for designing Smart Parking. The following papers helped us decide our project's algorithms and flow.

- In paper[1], authors state that car drivers and motorcycle riders spend a large amount of time finding an available parking space where slots are spread throughout multiple storeys which causes traffic congestion and long queues. The proposed system design described in the paper uses Python IDLE and the OpenCV library. OpenCV makes use of the combined edge detection and coordinates bound pixel sections in determining whether a parking space in the acquired footage is occupied or not. For the testing of the accuracy and reliability of the parking space identification system, sample videos of actual indoor parking garages were used. With this study, real-time image processing and updating of the parking slot availability offers an increased efficiency to the parking system and lower cost than installing individual car sensors in each parking space.

- In paper[2], the authors state that the smart parking occupancy detection system is a technology that aims to mitigate traffic congestion problems by reducing time for drivers to look for vacancy positions in car parking lots and providing efficient parking space utilization. Several reports show that the smart parking system not only alleviates traffic problems but also drives business growth and economic development within that neighborhood. In this paper, they have proposed a computer vision-based smart parking lot occupancy detection system employing low-complexity deep neural network architecture. A smart camera system that consists of a Raspberry Pi 3 attached to a camera utilizes a reduced-complexity deep neural network model to detect vacancy positions. They have trained and cross-validated their model using PKLot-Val dataset and tested the performance of their model using PKLot-Test and SWUpark datasets. SWUpark dataset has been created in the context of this research, accumulating visual information of parking lots at Srinakharinwirot University across several weather conditions. Through exhaustive hyperparameter tuning and stochastic gradient descent optimization, their model has achieved 88 percent accuracy, almost 15 percent higher than those obtained from the state-of-the-art approach.

- In paper[3], the authors state that People nowadays face the issue of finding empty or vacant slots for vehicle parking. For this, they spend nearly 6-8 minutes. The authors think that this is the reason which causes the major traffic in the big cities. For that, there are many facilities and techniques available in the market to reduce the tension of finding a vacant parking lot. Deep learning is one concept of data or image-processing that helps in this area. In Deep Learning various algorithms are used such as CNN, RCNN, MASKRCNN, YOLO, etc. The paper gives an elaborative comparison of available research on smart parking systems and talks about various algorithms with their results, pros, and cons. In this paper, they have deployed two algorithms on two datasets and measured their performance in terms of Precision score, Recall score, and F1 score. The authors concluded the paper by stating, the use of IoT with deep learning algorithms and giving a touch of wireless networking to the core system i.e. object detection and image processing will be more efficient in terms of result but somewhat not in terms of its overall cost. But using IoT, sensor selection will be a major task if one wants to build the system with a low budget. Still, problems like deem light, occlusion, and bad climate need to be addressed.

- In paper[4], the authors state that Parking management services need to achieve a variety of functions, such as parking information query, map route navigation, display, and guidance of spare parking spaces, intelligent payment, and car locating system, involving multidisciplinary technology, including edge computing, image processing technology, smartphone application development technology, and deep learning algorithm. While discussing algorithms they came to know that CNN's model can accurately identify parking spaces, but the real-time performance is poor. Hence their paper "Automatic Parking System Based on Improved Neural Network Algorithm and Intelligent Image Analysis" improves the convolutional neural network recognition model and establishes a one-time deep learning framework integrating the YOLO algorithm to improve the real-time performance of CNN model. In order to verify the superiority of the designed CNNs system, the simulation experiment is designed. The experimental comparison between CNNs and ZigBee and artificial parking was carried out.

- In paper[5], the authors state that designing an application for a parking system that provides not only the information about spaces for parking but also lets the user locate the slot to park vehicles so that the traffic can be reduced at the moment. The paper uses IoT in the above mentioned case. Further, to improve security the application uses computer vision to detect the number plate of vehicles using parking lots. This also helps the users to find their vehicle in case they forget where it was parked. In order to avoid the payment issue at entry or exit of the parking gate, the system also proposes a mobile payment process. This will also reduce the time consuming process in case a user faces issues related to physical money. The paper well defines each process step-wise. It also includes different requirements like functional, non-functional and technical. According to the paper, the application stores the data in the cloud. It further uses an Ultrasonic sensor to detect vehicles as a part of IoT. The system is embedded using Raspberry Pi. The paper has taken the real-time use case of Thailand's parking scenario.

- In paper[6], the authors state that India is one of the Countries with a High dense population. Due to this high population Transportation and Parking of Vehicles is the major issue faced by the people. This Paper aims to provide an Intelligent Parking System through Image Processing. The Image Processing Technique is used to identify the free empty Parking area to park the vehicles. In the proposed process the parking area can be marked with a certain specific number and a sensor and with the help of this sensor the empty space can be identified to park the vehicle. The Image processing Display consists of the seven segments of display in real-time. In addition to the display, they have implemented an Audio system in order to provide Oral information about the parking system. The seven-segment display can be used to identify the empty parking area with specific numbers. The specific numbers can be displayed in order to park the vehicle in a vacant position without any struggle. The proposed process is implemented in a software platform with the help of an Image processing technique and Hardware implementation can be done by interfacing with the Arduino Uno.

- In paper[7], the authors state that the Internet of Things (IoT) is able to connect billions of devices and services at any time in any place, with various applications. Recently, the IoT became an emerging technology. One of the most significant current research discussion topics on the IoT is about smart car parking. A modern urban city has over a million of cars on its roads but it does not have enough parking space. Moreover, most of contemporary researchers propose management of the data on the cloud. However, this method may be considered an issue since the raw data is sent promptly from distributed sensors to the parking area via the cloud and then received back after it is processed. This is considered an expensive technique in terms of data transmission as well as energy cost and consumption. While the majority of proposed solutions address the problem of finding unoccupied parking space and ignore some other critical issues such as information about the nearest car parking and road traffic congestion, this paper goes beyond and proposes an alternative method. The paper proposes a smart car parking system that will assist users to solve the issue of finding a parking space and to minimize the time spent searching for the nearest available car park. In addition, it provides users with road traffic congestion status. Moreover, the proposed system collects the raw data locally and extracts features by applying data filtering and fusion techniques to reduce the transmitted data over the network. After that, the transformed data is sent to the cloud for processing and evaluation by using machine learning algorithms.

- In paper[8], the authors state that nowadays the idea of Smart Cities had become very popular. With the advent of the Internet of Things, the concept of smart cities can be readily achievable. Extensive research is ongoing in the field of the Internet of Things to increase the quality of services offered in cities and to improve the productivity and reliability of urban infrastructure. The Internet of Things is addressing the most common problems faced in cities like the availability of car parking and traffic jams. This paper presents an Internet of Things-based Parking System for Smart Cities. The proposed parking system contains an IoT module deployed on-site for managing the available parking spaces. A platform is provided in the form of a portal for booking parking spaces.

- In paper[9], the authors state that traditional parking systems are very dependent on manual labor and possess multiple problems. A long time to find available parking slots, a big amount of cost to pay parking staff, and unavailability of parking data are some of the most common problems in this parking system. In this paper, they have proposed an Internet of Things (IoT) based smart parking system that has three main components, which are an application for parking management, an application for parking users, and an IoT platform. The IoT platform consists of two different granularities, which are in each parking slot and in each parking slot. IoT hardware in the parking slot will send information regarding the slot status, and IoT hardware in the parking lot will send information regarding any car that goes in or out. To reduce the number of sensors, crowdsensing from parking users and parking staff is used. This proposed design was implemented in the case study of X University. They conducted an evaluation of the implemented system and receive multiple feedback from parking system staff and parking users. Using the System Usability Scale, and received a score of 91 from the parking system staff and 80.14 from parking users. This evaluation result shows that the proposed IoT-based parking system is helpful and easy to use for parking staff and users

- In paper[10], the authors have explored the concept of the smart parking system and their categories. The classifications of various existing systems are explained. The parking system handles various technologies, and the categories of those techniques are given. The functions of the nodes in wireless sensor networks are classified.The parking system faces many problems in the parking environment. In order, to solve those problems, the smart parking system has been developed. Various approaches and researches are made to overcome the difficulties of parking area. As a result, many systems and technologies are developed for parking. The categories of various systems and technologies are explained in the following sections. The technologies of the parking system use wireless sensor network (WSN) for identification and communication process. In the WSN collection of nodes are deployed for various processes.

- In paper[11], the authors state that there is a tremendous increase in number of vehicles in last two decades. So, it becomes important to make effective use of technology to enable hassle free parking at public and/or private places. In traditional parking systems, drivers face difficulty in finding available parking slots. These systems ignore the fact of parking the vehicles on roads, time management in peak hours, wrong parking of a vehicle in a parking slot. To deal with above said issues, there is an urgent requirement of developing Smart Parking Systems. In this manuscript, the authors propose a Smart Parking System based on IoT and Machine learning techniques to answer the real time management of parking and uncertainties. The proposed solution utilizes smart sensors, cloud computing and cyber physical system. Development of graphical user interface for administrator and end-user is a major challenge as it requires to ensure smooth monitoring, control and security of parking system. Moreover, it needs to establish effortless coordination with an end-user. The proposed system is successful in smartly addressing the challenges such as indicating status of parking slot well in advance to end-user, use of reserved and unreserved parking slots, wrong parking, unauthorized parking, real time analysis of free and occupied slots, detecting multiple objects in a parking slot such as bike in car slot, fault detection in one or more components and traffic management during peak hours.

- In paper[12], the author state that this paper introduces a novel algorithm that increases the efficiency of the current cloud-based smart-parking system and develops a network architecture based on the Internet-of-Things technology. This paper proposed a system that helps users automatically find a free parking space at the least cost based on new performance metrics to calculate the user parking cost by considering the distance and the total number of free places in each car park. This cost will be used to offer a solution of finding an available parking space upon a request by the user and a solution of suggesting a new car park if the current car park is full. The simulation results show that the algorithm helps improve the probability of successful parking and minimizes the user waiting time. We also successfully implemented the proposed system in the real world.

Chapter 3

Project Design

3.1 System Design

The project offers an advanced solution for voluntary parking places by upgrading a parking management system that uses machine learning to enhance the environment for vision-based vacant parking space detection. The main aim is to have the maximum number of vehicles that may be parked in a systematic manner in the vacant lot.

This system intends to locate and apprehend the real-time vacant parking area. The system contains a digital camera installed on the rooftop of our college basement or a few assisting poles at a positive angle where it covers most places of the vehicle parking zone which is being used for taking the input. The images received from the camera are then fed to the processing module, which detects the area of interest which includes the place to be included for parking areas. A vehicle detection Module is used to locate the vehicles inside the basement using YoloV5. This module tracks and detects the parking area in an image. The parking area detection module generates the bounding boxes for parking so it can be seen by a person on a vehicle parking app.

Whenever the user opens the app, he will see the virtual representation of the parking area and the number of vacant areas in the vehicle parking zone and reserve a parking spot when the driver is at a particular distance from the college area. To offer ease to the person; an assist display screen will be provided which will inform the person how to check the status of the parking lot.

Furthermore, the users who do not use our application and directly enter the college basement looking for a vacant spot to park their vehicle will have an OLED dashboard displayed at the entrance stating the number of vacant spots available. If no vacant spots are available then the dashboard will display zero or full parking as no spots are available.

Python Programming language version 3.9.13 is used to train the models and do the processing. Mobile Application for Users and Admin is built using Java and Android Studio. Cloud will be used for deploying our system and Message Queuing Telemetry Transport connection protocol will be used for creating the connection between IoT devices and the dashboard.

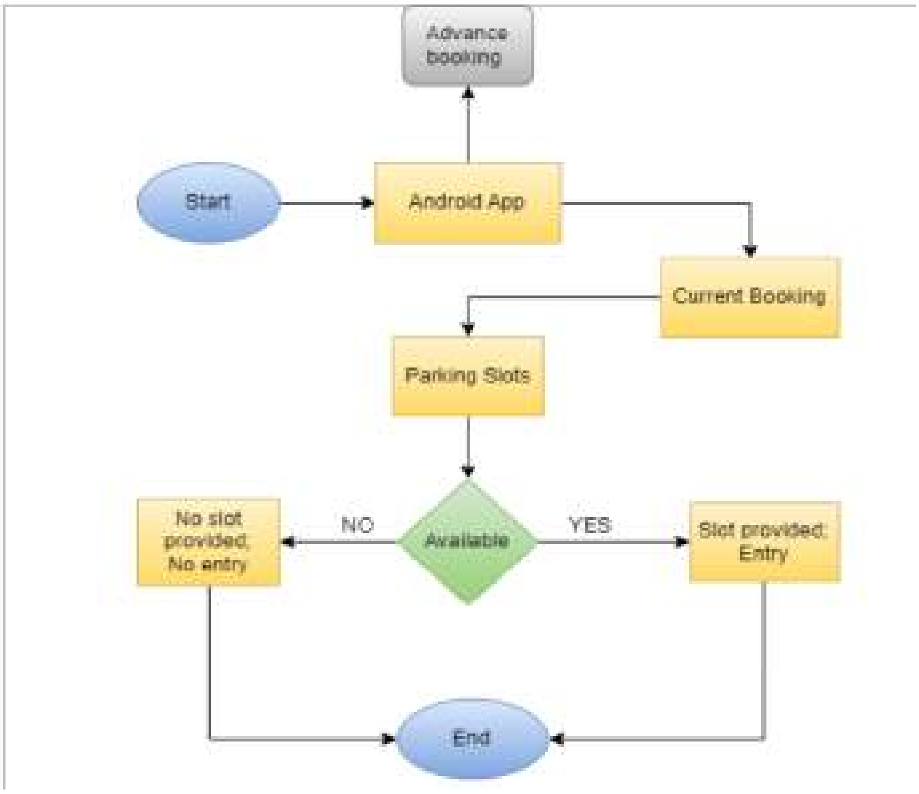


Figure 3.1: System Design

3.2 Mathematical Formulas

YOLOv5 is a popular object detection algorithm that uses convolutional neural networks (CNNs) for real-time object detection in images. It uses a single neural network to process an entire image. The image is divided into regions and the algorithm predicts probabilities and bounding boxes for each region. While it is not directly associated with a mathematical equation, it is based on several mathematical concepts and techniques used in deep learning and computer vision.YOLO is a deep network, it uses residual and dense blocks in order to enable the flow of information to the deepest layers and to overcome the vanishing gradient problem. However one of the perks of using dense and residual blocks is the problem of redundant gradients. One of the key components of YOLOv5 is the bounding box prediction, which involves estimating the coordinates of the bounding boxes around detected objects. The coordinates of a bounding box can be represented as (x, y, w, h), where (x, y) are the coordinates of the box's top-left corner, and (w, h) are the width and height of the box, respectively.
Here are the main formulas used in YOLOv5:

Bounding Box Prediction:

The predicted bounding box coordinates (x, y, w, h) for an object in YOLOv5 are calculated as follows:

- x = (sigmoid(tx) + cx) * stride
- y = (sigmoid(ty) + cy) * stride
- w = pw * exp(tw)
- h = ph * exp(th)

where:

- sigmoid() is the sigmoid activation function that maps the predicted values to the range [0, 1].
- tx, ty, tw, th are the predicted values of the bounding box coordinates.
- cx, cy are the coordinates of the cell in the grid where the object is detected.
- pw, ph is the anchor box dimensions used for prediction.
- exp() is the exponential function.
- stride is the downscaling factor of the input image during the network's feature extraction process, used to map the predicted bounding box coordinates back to the original image space.

Object Confidence Prediction:

The predicted object confidence score (objectness score) for an object in YOLOv5 is calculated as follows:

- objectconfidence = sigmoid(tobjectconfidence)

  where:

- tobjectconfidence is the predicted value of the object confidence score.

Class Prediction: The predicted class probabilities for an object in YOLOv5 are calculated using softmax activation function as follows:

- classprobabilities = softmax(tclassprobabilities)

  where:

- tclassprobabilities is the predicted vector of class probabilities.

These formulas are used in YOLOv5's output layer to predict the class labels, bounding box coordinates, and object confidences for detected objects in an image. The network is trained using labeled data to learn the optimal values for the predicted parameters during the training process.

3.3 Use Case diagram

A use case diagram is a type of behavioral diagram in UML (Unified Modeling Language) that depicts the interactions between a system and its users or external systems. It provides a high-level overview of the functional requirements of a system and the different actors or users that interact with it.

Use case diagrams are useful for communicating the overall functionality of a system and can help identify potential errors or omissions in the system design. They can also aid in the identification of various actors and their roles, which can assist in defining the system's requirements and design. Use case diagrams are often used during the requirements gathering and analysis phase of software development projects.

The Use Case Diagram below shows different entities of the application. The relation between users and what they can do with the system. The classification of spots: vacant and occupied would be evaluated and the output will be containing the count of occupied and vacant spots from images. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements.

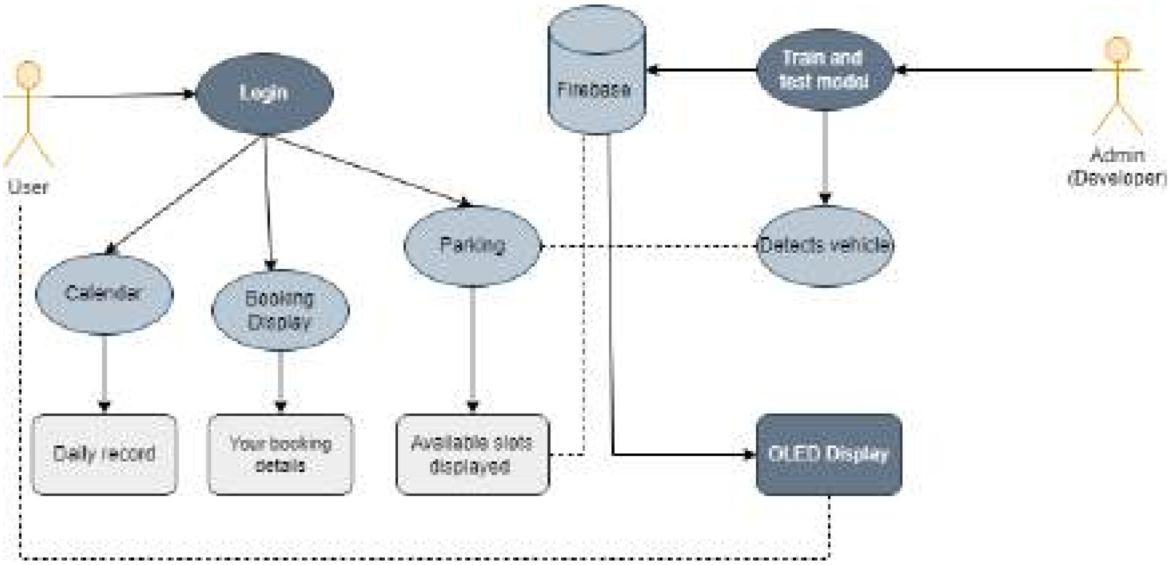


Figure 3.2: Use Case Diagram

3.4 Sequence diagram

A sequence diagram is a type of interaction diagram in UML (Unified Modeling Language) that shows the interactions between objects or components in a system over time. It describes how objects collaborate with each other to achieve a specific task or functionality.

In a sequence diagram, objects are represented by lifelines, and the interactions between them are shown by the arrows that move from one lifeline to another. Each arrow represents a message being passed between the objects, and the sequence in which the messages are exchanged is shown on the vertical axis of the diagram.

The sequence diagram below shows how objects operate with one another and in what order in our system. First, the input image data would be pre-processed using yolo preprocessor and coefficients would be extracted which would be sent for classification to the Yolov5 detection module. The higher the precision, and recall coefficients greater the accuracy, The classification of three types of cries: vacant and occupied would be evaluated and the output will be containing the count of spots based on whether they are occupied or not.

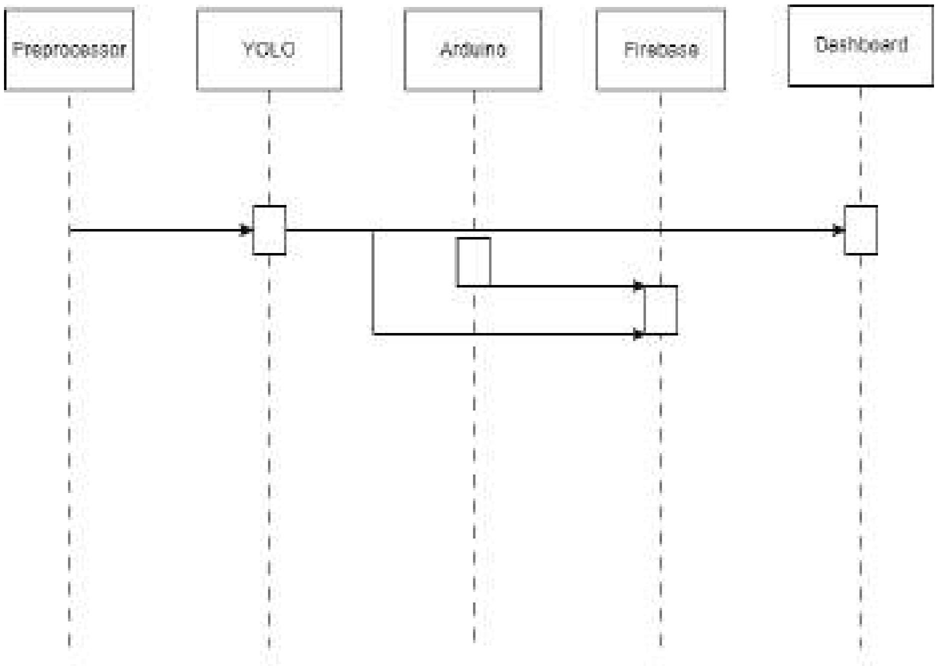


Figure 3.3: Sequence Diagram

Chapter 4

Project Implementation

4.1 Dataset

Custom datasets were created by gathering, organizing, and annotating data that is relevant and specific to our college parking area. These datasets were meticulously crafted to meet the specific needs of a machine learning model yolov5, allowing for targeted training and evaluation. The process of creating a custom dataset involves thorough data preprocessing, including data cleaning, normalization, and labeling to ensure accuracy and consistency. Here are a few images that were captured and annotated.

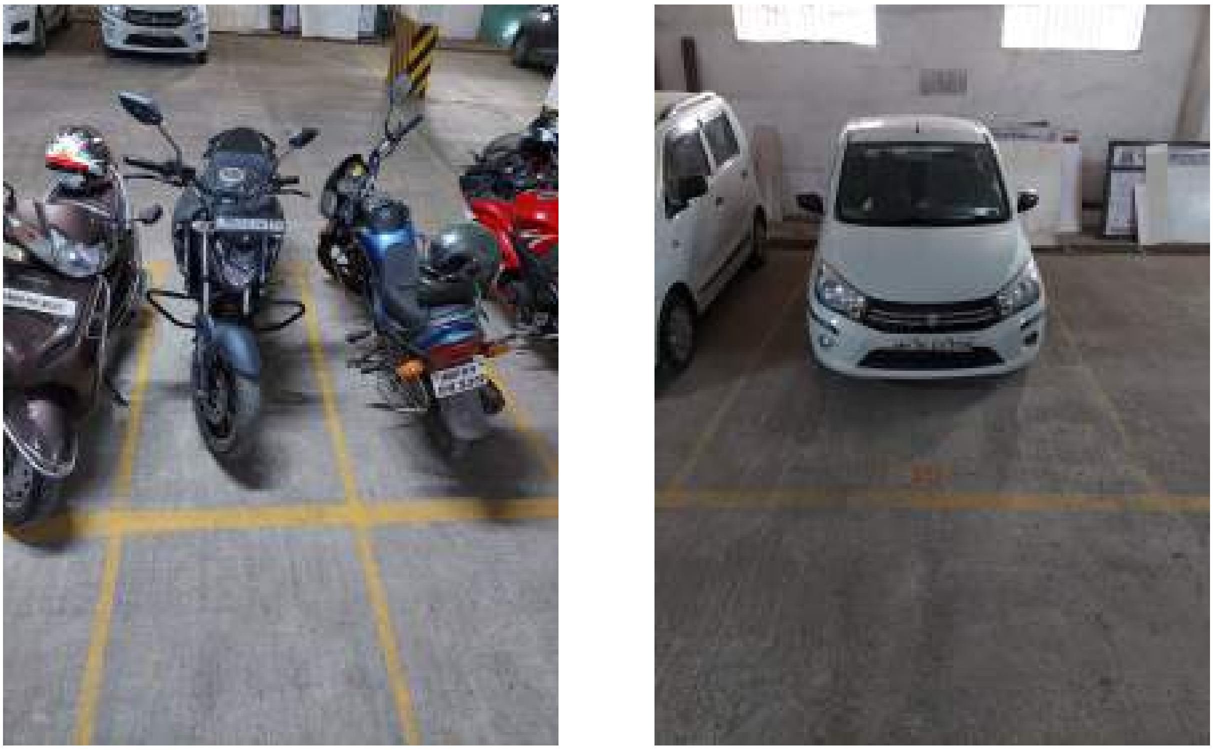


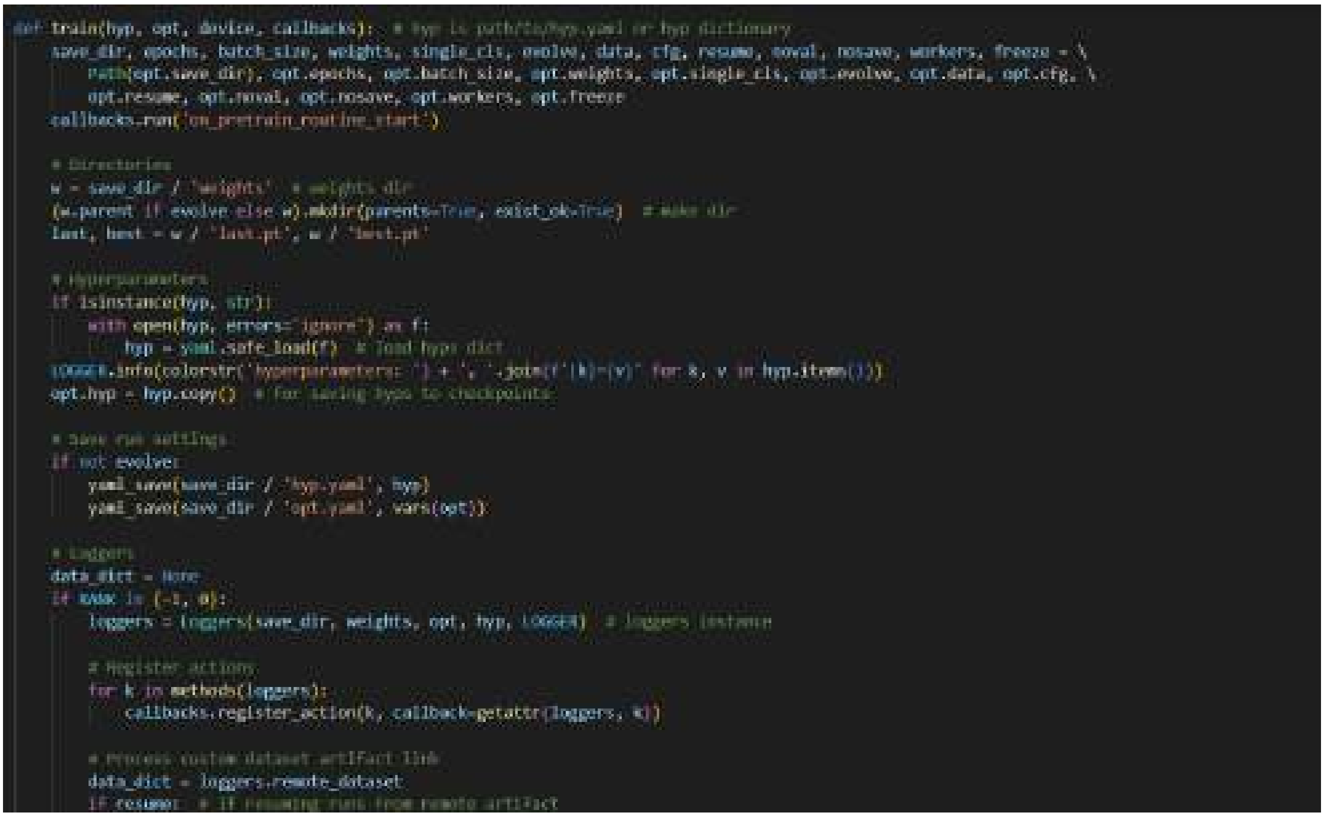
Figure 4.1: Sample Training Data

4.2 Code Snippets

4.2.1 Model Training

The training file in YOLO contains the annotations and labels for the training data, which are used to train the neural network model. The YOLO training file typically consists of two main parts: the image data and the corresponding annotation data. The image data includes the actual images that are used for training, which are typically in a standardized format such as JPEG or PNG. The annotation data contains information about the objects in the images, including their bounding box coordinates, class labels, and optionally, additional attributes such as object poses or orientations.

The following is the train function that is used to train the YOLOv5 model and store the result in .pt file format.

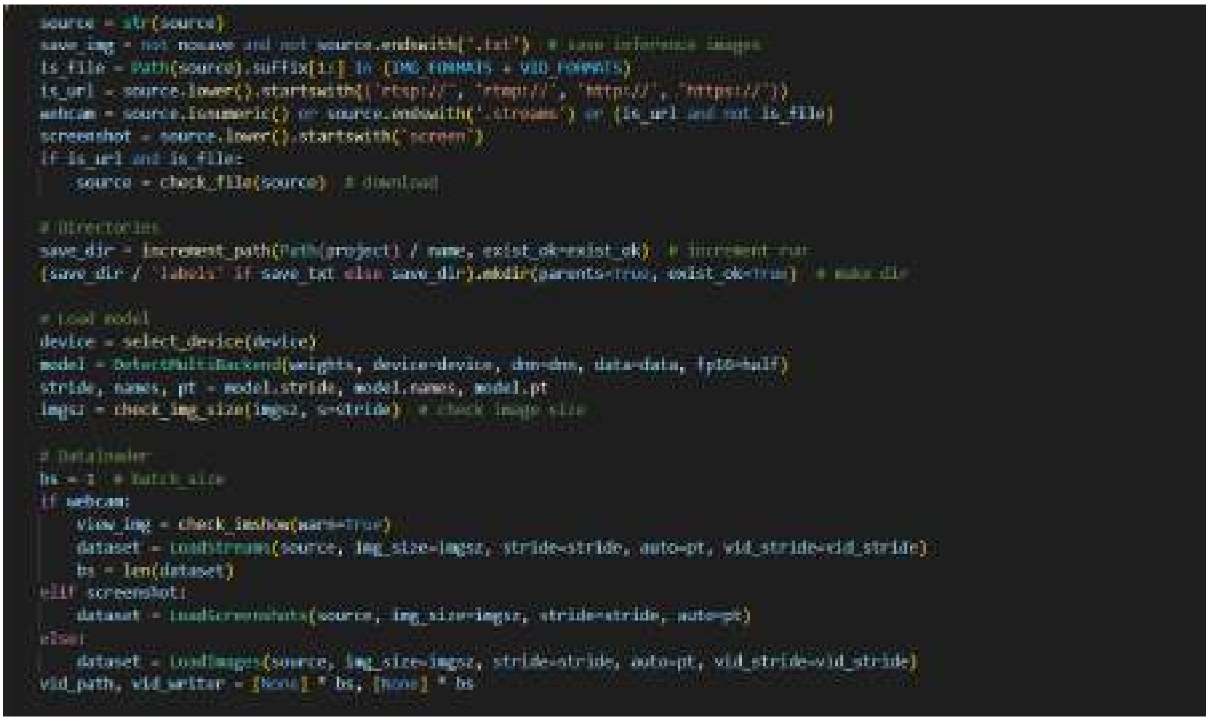


```
def train(hyp, opt, device, callbacks):  # hyp is path/to/hyp.yaml or hyp dictionary
    save_dir, epochs, batch_size, weights, single_cls, evolve, data, cfg, resume, noval, nosave, workers, freeze = \
        Path(opt.save_dir), opt.epochs, opt.batch_size, opt.weights, opt.single_cls, opt.evolve, opt.data, opt.cfg, \
        opt.resume, opt.noval, opt.nosave, opt.workers, opt.freeze
    callbacks.run('on_pretrain_routine_start')

    # Directories
    w = save_dir / 'weights'  # weights dir
    (w.parent if evolve else w).mkdir(parents=True, exist_ok=True)  # make dir
    last, best = w / 'last.pt', w / 'best.pt'

    # Hyperparameters
    if isinstance(hyp, str):
        with open(hyp, errors='ignore') as f:
            hyp = yaml.safe_load(f)  # load hyps dict
    LOGGER.info(colorstr('hyperparameters: ') + ', '.join(f'{k}={v}' for k, v in hyp.items()))
    opt.hyp = hyp.copy()  # for saving hyps to checkpoints

    # Save run settings
    if not evolve:
        yaml_save(save_dir / 'hyp.yaml', hyp)
        yaml_save(save_dir / 'opt.yaml', vars(opt))

    # Loggers
    data_dict = None
    if RANK in {-1, 0}:
        loggers = Loggers(save_dir, weights, opt, hyp, LOGGER)  # loggers instance

        # Register actions
        for k in methods(loggers):
            callbacks.register_action(k, callback=getattr(loggers, k))

        # Process custom dataset artifact link
        data_dict = loggers.remote_dataset
        if resume:  # If resuming runs from remote artifact
```

Figure 4.2: Model Training 1

4.2.2 Detection

The detection file in YOLO contains the pre-trained weights and configurations of the neural network model, which are used to detect objects in new images or videos. The YOLO detection file typically includes two main components: the network architecture and the pre-trained weights. The network architecture defines the structure and configuration of the neural network, including the number of layers, their types (such as convolutional, pooling, and fully connected), and their connections. The pre-trained weights are the learned parameters of the neural network that were obtained during the training process using a labeled dataset



```
source = str(source)
save_img = not nosave and not source.endswith('.txt')  # save inference images
is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
webcam = source.isnumeric() or source.endswith('.streams') or (is_url and not is_file)
screenshot = source.lower().startswith('screen')
if is_url and is_file:
    source = check_file(source)  # download

# Directories
save_dir = increment_path(Path(project) / name, exist_ok=exist_ok)  # increment run
(save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True)  # make dir

# Load model
device = select_device(device)
model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
stride, names, pt = model.stride, model.names, model.pt
imgsz = check_img_size(imgsz, s=stride)  # check image size

# Dataloader
bs = 1  # batch_size
if webcam:
    view_img = check_imshow(warn=True)
    dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
    bs = len(dataset)
elif screenshot:
    dataset = LoadScreenshots(source, img_size=imgsz, stride=stride, auto=pt)
else:
    dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
vid_path, vid_writer = [None] * bs, [None] * bs
```

Figure 4.3: Detection

4.3 IoT Connections

IoT (Internet of Things) is used in parking systems to enhance efficiency, convenience, and management of parking spaces. Iot is used to display the count of available and occupied spots in our college basement through OLED.

4.3.1 OLED

An organic light-emitting diode, also known as an organic electroluminescent diode, is a light-emitting diode in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current. Once the count of vacant parking spots from Firebase is retrieved, update the OLED display with the retrieved data. The OLED display library for your specific hardware platform displays the count of vacant parking spots on the OLED screen in a visually appealing way, such as using text, numbers, or graphical elements is used.

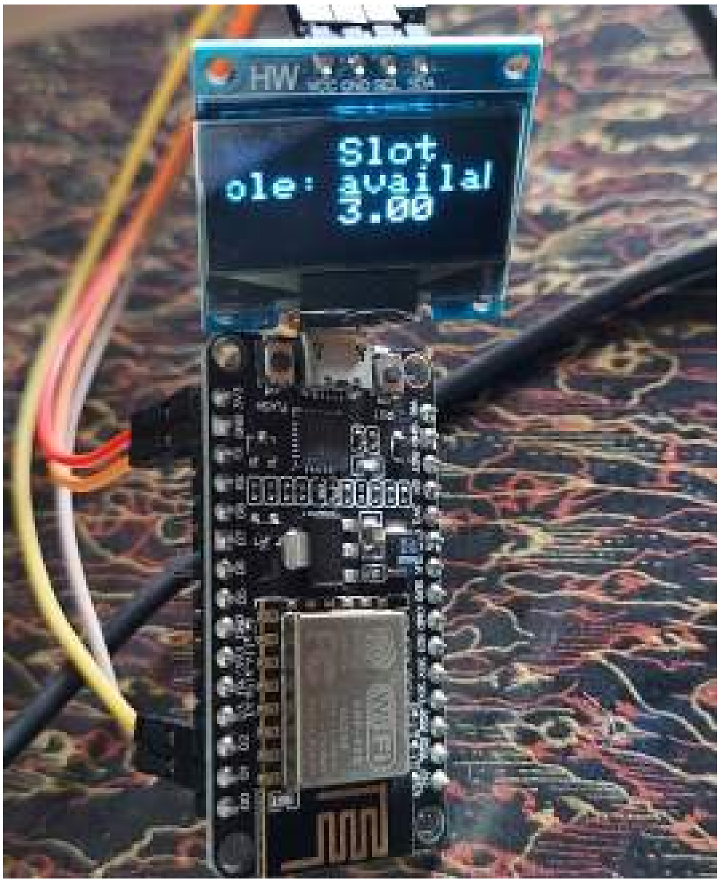
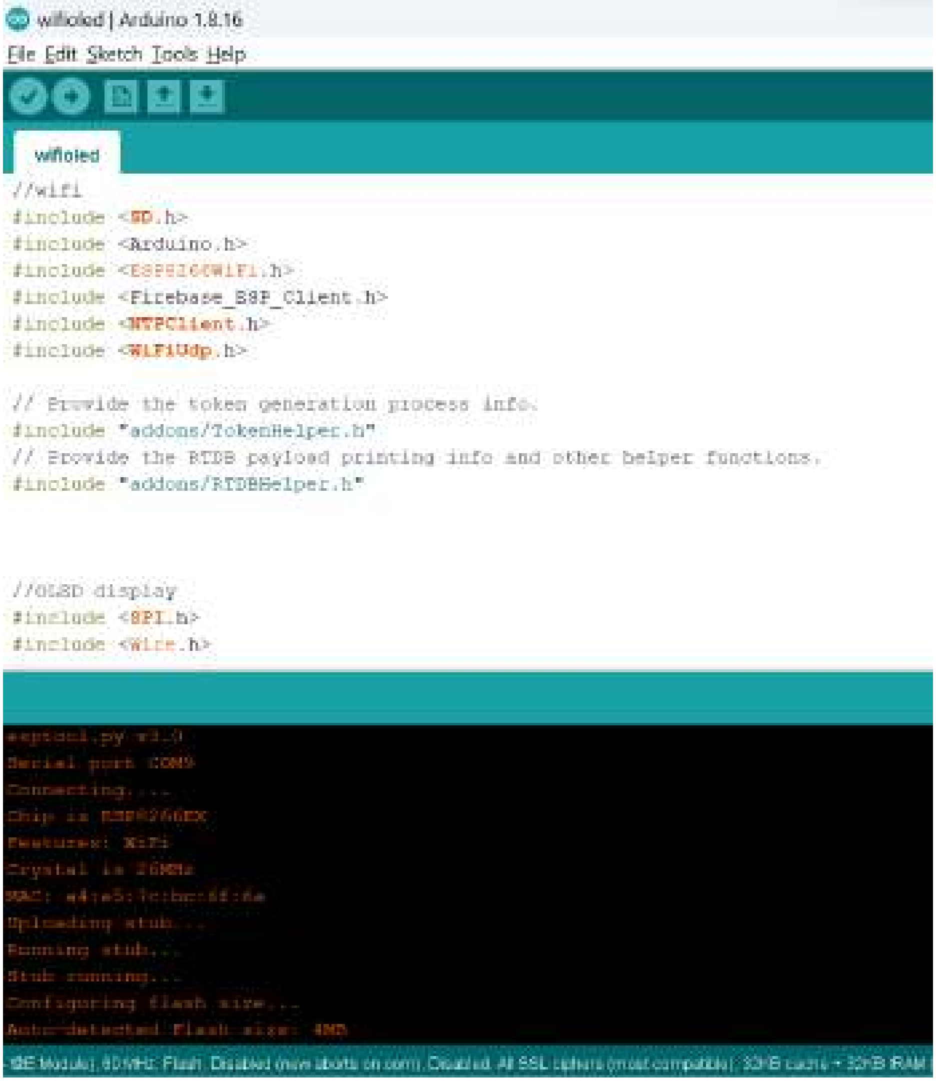


Figure 4.4: Count display using OLED

4.3.2 Arduino Interface

Arduino is an open-source hardware platform that provides a simple and accessible way to interface with various electronic components and sensors. Arduino allows you to create interactive projects by writing and uploading code to a microcontroller board that is connected to your computer or other devices. To display the count of vacant or available parking spots in real-time, we have set up a listener using the Firebase Arduino library to listen for changes in the Firebase Realtime Database. Whenever the count of vacant or available parking spots changes in the database, the updated count is displayed on the OLED in real-time.



```
//wifi
#include <SD.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

// Provide the token generation process info.
#include "addons/TokenHelper.h"
// Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"



//OLED display
#include <SPI.h>
#include <Wire.h>
```

```
esptool.py v3.0
Serial port COM9
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: e4:e5:7c:bc:6f:6e
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
```

-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (

Figure 4.5: Arduino interface

4.4 User Interface

Users can log in by entering his/her email and password and the new user can sign up by clicking on register. Users can navigate through the app and book the spot through the app and also can see which spots are available for parking. Once the booking is successful an alert is displayed

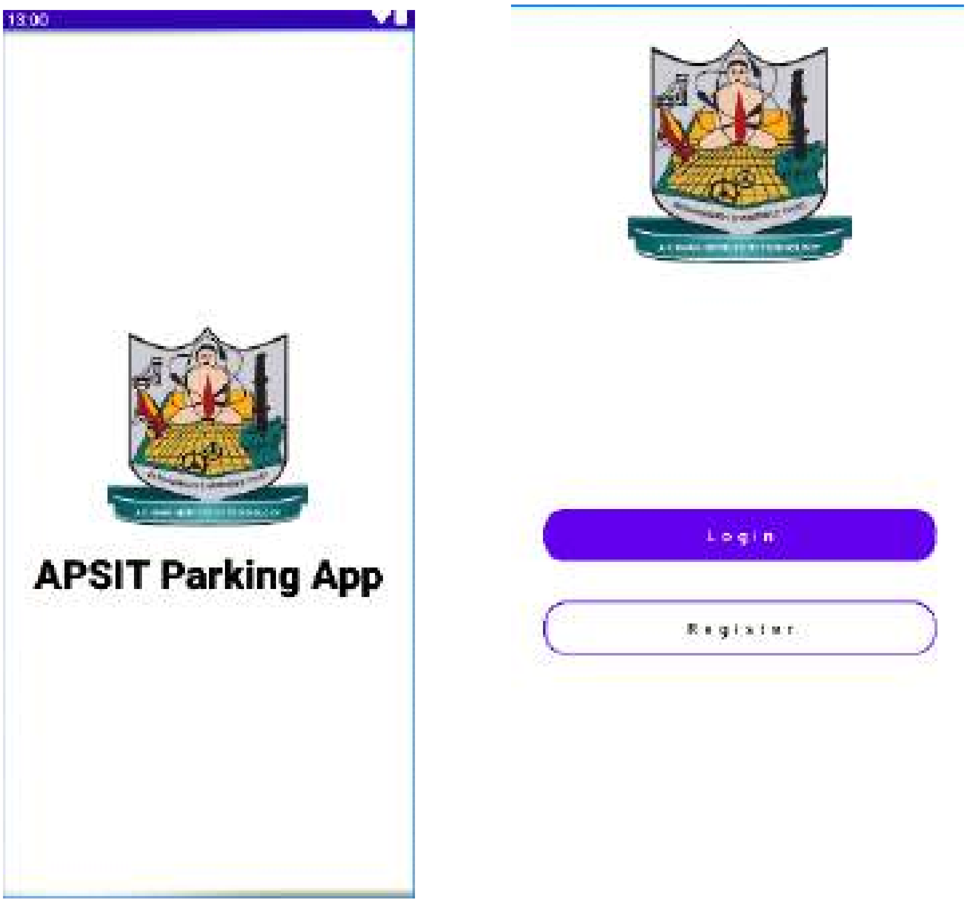


Figure 4.6: Parking App Launching Page and Log in/Register Page

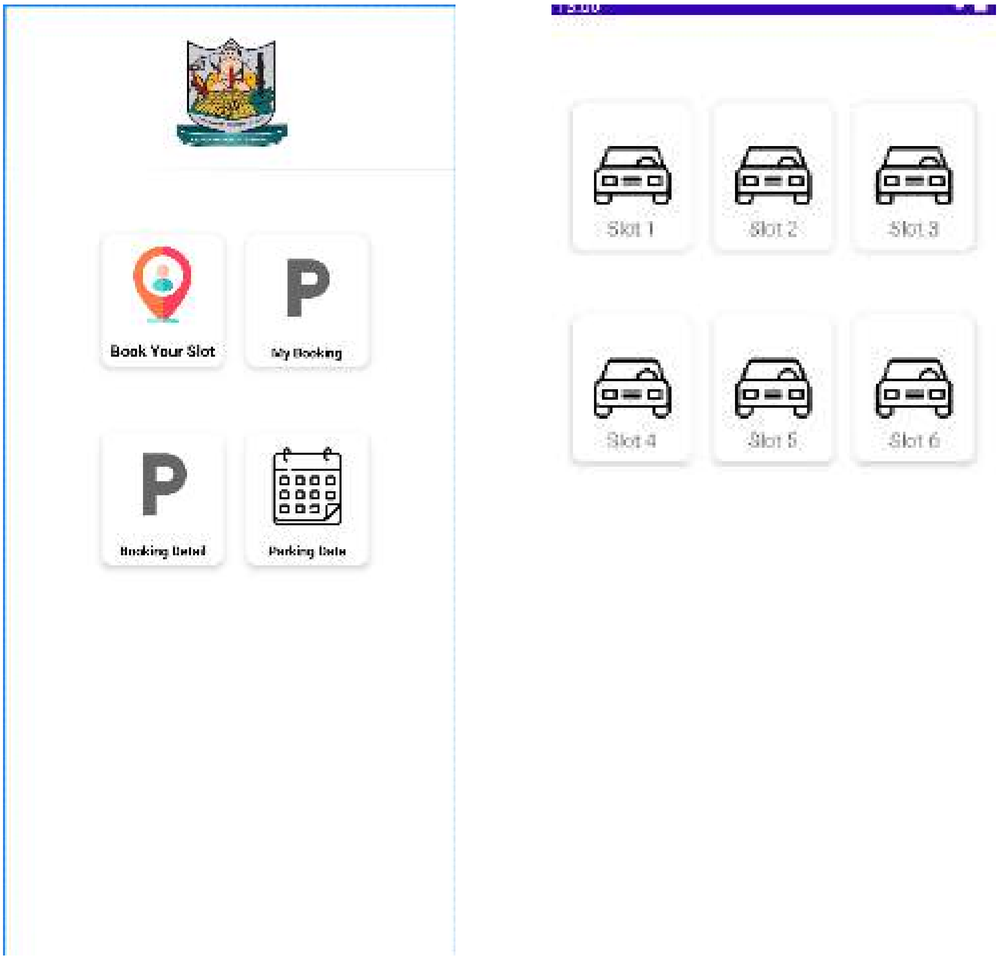


Figure 4.7: Home Page and Slot display in grid view

The above figure displays the pages that will be displayed to a user once he logs in to the app using his Login credentials.

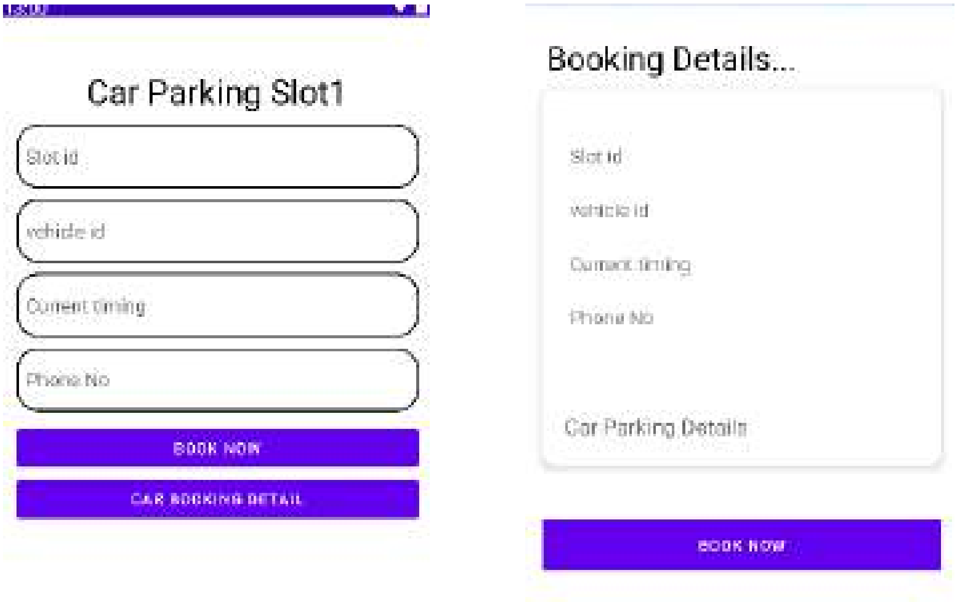


Figure 4.8: Slot Booking Page and Booing Detail Page

The above figure displays the pages where a user can check for available slots and make a booking.

Chapter 5

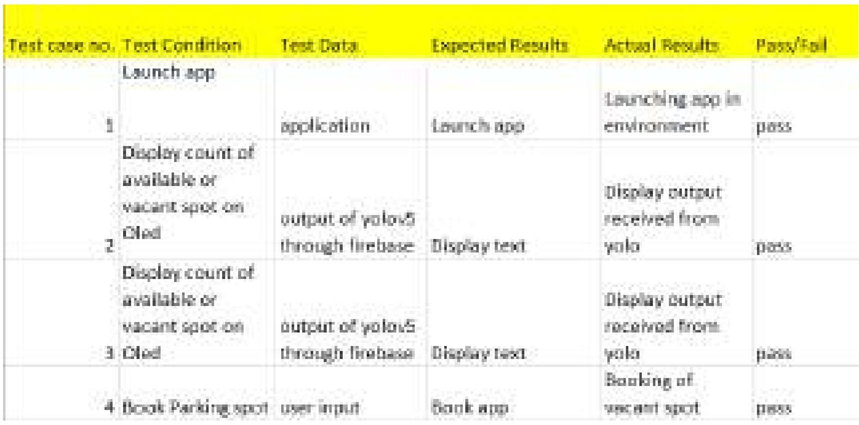
Testing

5.1 Functional Testing

5.1.1 Unit Testing

Unit testing is the first level of testing, which is typically performed by the developers-themselves. At the code level, it is the process of ensuring that individual components of software are functional and work as intended. Unit testing can be done manually, however automating the process will reduce delivery times and boost test coverage. Because flaws will be detected earlier in the testing process and will take less time to fix than if they were discovered later, debugging will be easier as a result of unit testing. It helped us understand the desired output of each module, which we had broken down into separate units. It helped us in classifying the cry categories on the basis of algorithm that we have used. The main objective of unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

5.1.2 Various TestCases



Test case no.	Test Condition	Test Data	Expected Results	Actual Results	Pass/Fail
1	Launch app	application	Launch app	Launching app in environment	pass
2	Display count of available or vacant spot on Oled	output of yolov5 through firebase	Display text	Display output received from yolo	pass
3	Display count of available or vacant spot on Oled	output of yolov5 through firebase	Display text	Display output received from yolo	pass
4	Book Parking spot	user input	Book app	Booking of vacant spot	pass

Figure 5.1: Testcases

Chapter 6

Result

6.1 Labels

The Yolov5 model was trained with 2 class labels namely occupied and vacant. The following are the images of the result of the model trained.The model learns to predict bounding boxes and class probabilities for objects in new images based on the patterns it learns from the labeled training data. During training, the model adjusts its parameters to minimize the difference between its predictions and the ground truth labels, using techniques such as backpropagation and gradient descent.

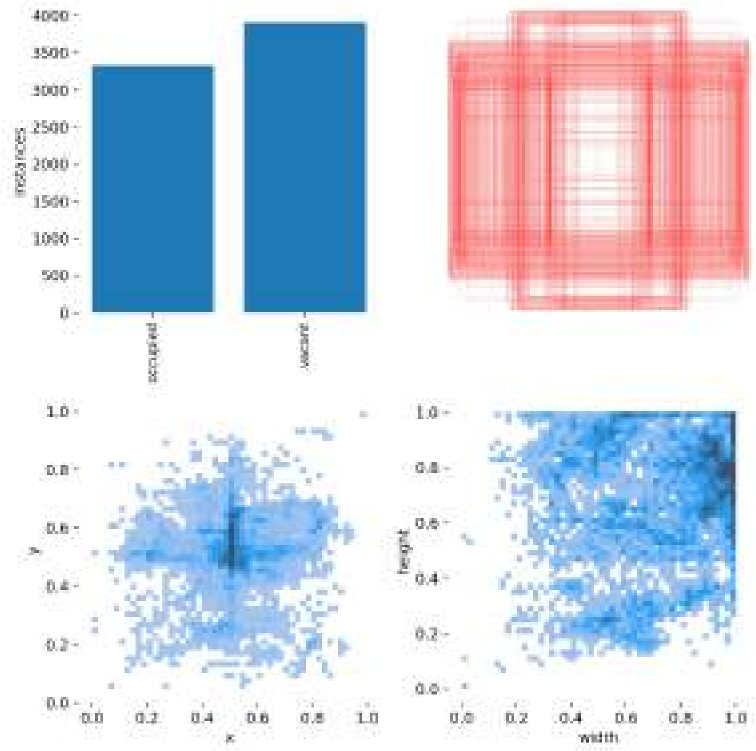


Figure 6.1: Labels

6.2 Precison Recall Curve

The Precision-Recall (PR) curve is a graphical representation of the trade-off between precision and recall in the context of object detection using YOLOv5. It is a useful tool for evaluating the performance of the model and determining an appropriate threshold for object detection.

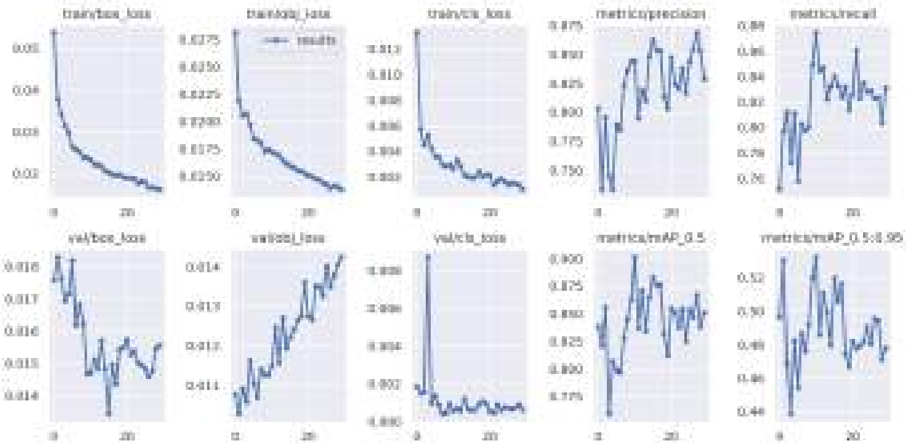


Figure 6.2: Precision-Recall-Accuracy Curve

6.3 Confusion Matrix

In YOLOv5, a confusion matrix is a performance evaluation tool that provides a tabular representation of the model's predictions and the ground truth labels during object detection tasks. It is commonly used to assess the accuracy and effectiveness of the model's predictions and identify any misclassifications or false positives/negatives.

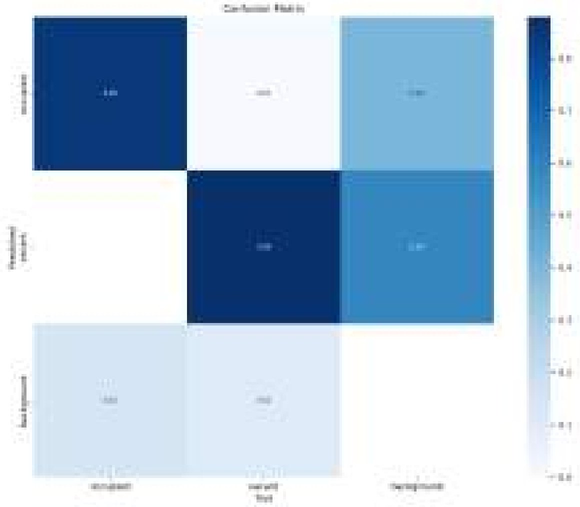


Figure 6.3: Confusion Matrix

Chapter 7

Conclusion

The model was trained using the YoloV5 algorithm on a custom dataset with an accuracy of 90 percent as per the evaluation in the stipulated time. The proposed design of a smart parking system based on object detection has been successfully tested and executed using multiple indoor parking area images. The user interface was developed using Java and Android Studio. The user can get a real-time update on the parking lot to determine whether there are any open spots for them to park their automobile.

If the proposed architecture is Implemented in the day to day life the parking of vehicle can be made easier. The Process of identifying the parking area and the number of empty parking slots can be determined with the help of an Image processing technique. The parking slots can be easily identified and the shape of that particular slot can be determined which results in occurrence of parking the vehicle in safe area within short span of time without any delay.

Future Scope

The Proposed design is implemented for a small area of 6 Parking slots in a particular region.

- The proposed idea can be extended for all around the city by providing additional information through the GPS module.
- The parking area can be identified in a particular zone through GSM with a Mobile application.
- The authencity of the booking can be verified and validated by implementing QR code system.

Bibliography

[1] Chyn Ira C. Crisostomo, Royce Val C. Malalis, Romel S. Saysay and Renann G. Bal-Dovino, “A Multi-storey Garage Smart Parking System based on Image Processing”, IEEE Xplore, 2019-in press.

[2] Chantri Polprasert, Chaiyaboon Sruayiam, Prathan Pisawongprakan and Sirapob Ter Avetchakarn, “A Camera-based Smart Parking System Employing Low-complexity Deep Learning for Outdoor Environments”, IEEE Xplore, 2020-in press.

[3] Adesh Pawar, Ajay Pawar, Ashish Pawar, Ganesh Pawar and Prof. Anagha Chaudhari, “An Elaborative Study of Smart Parking Systems”, IJERT, Department of Computer Engineering Pimpri Chinchwad College of Engineering, Vol. 10 Issue 10, October-2021.

[4] Yucheng Guo and Hongtao Shi, “Automatic Parking System Based on Improved Neural Network Algorithm and Intelligent Image Analysis”, Computational Intelligence and Neuroscience, Volume 2021, Article ID 4391864, 2021.

[5] Rachapol Lookmuang, Krit Nambut, and Sasiporn Usanavasin, “Smart Parking Using IoT Technology”, International Conference on Business and Industrial Research (ICBIR), Thailand, August 2018.

[6] K. Kaarthik, a. sridevi, and c. vivek, “Image processing based intelligent parking system,” 2017 IEEE Xplore, International Conference On Electrical, Instrumentation and Communication Engineering (ICEICE), 2017.

[7] Wael Alsafery, Badraddin Alturki, Stephan Reiff-Marganiec and Kamal Jambi, “Smart Car Parking System Solution for the Internet of Things in Smart Cities”, IEEE Xplore, United Kingdom, 2018.

[8] D.Vakula and Yeshwanth Krishna Kolli, “Low Cost Smart Parking System for Smart Cities”, National Institute of Technology, Warangal, Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS), IEEE Xplore, 2017.

[9] Ratnadira Widyasari, Muhammad Zuhri Catur Candra, and Saiful Akbar, “IoT-based Smart Parking System Development”, Institut Teknologi Bandung Bandung, Indonesia, International Conference on Data and Software Engineering (ICoDSE), IEEE Xplore, 2019.

[10] G. Revathi, V. R. Sarma Dhulipala, “Smart Parking Systems and Sensors: A Survey”, Anna University of Technology Tiruchirappalli, Tamil Nadu, IEEE Xplore, 2012.

[11] Ravi Kumar and Gupta, Ravi Kumar and Rani, Geeta, Machine Learning and IoT based Real Time Parking System: Challenges and Implementation Proceedings of the International Conference on Innovative Computing Communications (ICICC) 2020.

[12] THANH NAM PHAM1, MING-FONG TSAI1, DUC BINH NGUYEN1,CHYI-REN DOW1, AND DER-JIUNN DENG2,"A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies", SPECIAL SECTION ON EMERGING CLOUD-BASED WIRELESS COMMUNICATIONS AND NETWORKS,2015.

[13] Siou-Hong Liou, Yun-Chin Hsieh, and Cheng-Yuan Chang, "Design and Implementation of A Smart Parking Management System for Smart Cities ", International Conference on Consumer Electronics-Taiwan (ICCE-TW), IEEE Xplore, 2018.

[14] Dharmini Kanteti, D V S Srikar and T K Ramesh, "Smart Parking System For Commercial Stretch In Cities", International Conference on Communication and Signal Processing, IEEE Xplore, April 2018.

[15] Dharmini Kanteti, D V S Srikar, and T K Ramesh, "Intelligent Smart Parking Algorithm", Amrita University, Bengaluru, IEEE Xplore, 2017

Appendices

Appendix-I: Python Libraries

- pip install opencv-python
- pip install numpy
- pip install matplotlib
- pip install torch
- pip install yolov5
- pip install tensorboad=2.4.1
- pip install clearml=1.2.0
- pip install pandas=1.1.4
- pip install seaborn=0.11.0
- pip install coremltool=6.0
- pip install onnx=1.12.0
- pip install onnx-simplifier=0.4.1
- pip install nvidia-pyindex
- pip install nvidia-tensorrt
- pip install scikit-lear=1.1.2
- pip install tensorflow=2.4.1
- pip install tensorflow=3.9.0
- pip install openvino-dev
- pip install firebase-admin
- pip install user firebase-admin

Publication

Paper entitled **“IoT and ML based Cross Platform Application for Designing Smart Parking”** is selected to be presented at **“International Conference on Contemporary Challenges in Science and Engineering Applications 2023 (IC3SEA 2023)”** by **”Snehal Shanbhag, Pranjali Shimpi, Akansha Rawat, Prof. Sonal Jain, Prof. Charul Singh and Dr. Kiran Deshpande”** and will be published in Xplore and CSDL.