

# **SMARTPARK-X USING MACHINE LEARNING**

## **Major Project Report**

Submitted in partial fulfillment for the award of the degree of

## **BACHELOR OF TECHNOLOGY**

**In**

## **COMPUTER SCIENCE and ENGINEERING**

**By**

YATHESWAR ETHALAPAKA	19L31A0546
TRISULE KUMAR REDDY SEELAM	19L31A0583
CHAKRADHAR KAMMINANA	19L31A0504
PAVAN KALYAN MAHANTY	20L35A0502

**Under the Guidance of**

**Mrs. Srilatha Yalamati**

**ASSISTANT PROFESSOR**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (Autonomous)**

Affiliated to JNTU Kakinada & Approved by AICTE, New Delhi

Re-Accredited by NBA & NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2015, ISO 14001:2015, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM – 530 049

**VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY**

**Department of Computer Science and Engineering**



## CERTIFICATE

This is to certify that the project report entitled “**SmartParkX**” the bona fide record of project work carried out under my supervision by **Yatheswar Ethalapaka (19L31A0546), Trisule Kumar Reddy Seelam(19L31A0583), Chakradhar Kamminana (19L31A0504), and Pavan Kalyan Mahanty(20L35A0502)**, during the academic year 2019-2023, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University, Kakinada. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

**Head of the Department**

Mr. B. Dinesh Reddy

ASSOCIATE PROFESSOR

**Signature of Project Guide**

Mrs. Srilatha Yalamati

ASSISTANT PROFESSOR

**EXTERNAL EXAMINER**

## DECLARATION

We hereby declare that the project report entitled “**SmartParkX**”, has been written by us and has not been submitted either in part or whole for the award of any degree, diploma or any other similar title to this or any other university.

YATHESWAR ETHALAPAKA	19L31A0546
TRISULE KUMAR REDDYSEELAM	19L31A0583
CHAKRADHAR KAMMINANA	19L31A0504
PAVAN KALYAN MAHANTY	20L35A0502

Date:

Place: Visakhapatnam

## ACKNOWLEDGEMENT

We consider it a privilege to thank all those people who helped us a lot for successful completion of the project “**SmartParkX**”.

First of all, we would like to thank our project guide **Mrs. SriLatha Godthi, Assistant Professor, Department of Computer Science & Engineering** for helping us a lot in completing our project work and for enlightening us with constructive suggestions for solving our problems patiently and helping us to improve the quality of work.

We would like to thank our ever-inspiring **Head of the Department of Computer Science and Engineering, Mr. B. Dinesh Reddy**, for his spontaneous response to every request though he is busy with his hectic schedule of administration and teaching.

We would like to acknowledge with much appreciation the crucial role of our **Project Coordinator, Mr. Ch. Sekhar, Associate Professor, Department of Computer Science & Engineering** for helping us a lot in completing our project through the academic year. We just wanted to say thank you for being such a wonderful educator as well as a person.

We would like to thank **Principal** of Vignan’s Institute of Information Technology **Dr. B. Arundhati**, for her encouragement to us during the course of this project work.

We also express our sincere thanks to our beloved **Chairman** of Vignan’s Institutions **Dr. L. Rathaiah**, who has given us a lot of support and freedom during our project work.

We thank the entire **faculty** who have been a constant source of support during our study tenure.

## ABSTRACT

There has been a tremendous increase in the number of vehicles in the 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 spaces. These systems ignore the fact of parking vehicles on roads, time management in peak hours, wrong parking of a vehicle in a parking slot. Moreover, the traditional systems require more human intervention in a parking zone. To deal with above said issues, there is an urgent requirement to develop Smart Parking Systems. In this **SmartParkX**, we have proposed a Smart Parking System based on Machine learning techniques to answer the real time management of parking and uncertainties. The proposed solution utilizes Machine Learning models, Cloud Database, API's, Messaging, Payments and physical systems. Development of a graphical user interface for an end-user is a major challenge as it requires smooth monitoring, control and security of the 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. The system minimizes human intervention and saves time, money and energy.

**Key Words:** Machine Learning, YOLOv3, Non-Maximum Suppression (NMS), Automatic Number Plate Recognition (ANPR), Active Contour Model, Smart Parking, Space Counter, Computer vision, Cloud DB, Payments, Messaging, API's.

# CONTENTS

<b>TITLE</b>	<b>PAGE NO</b>
Declaration	i
Acknowledgement	ii
Abstract	iii
List of Figures	iv

<b>TITLES</b>	<b>PAGE NO</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>1.1 Machine Learning</b>	<b>2</b>
1.1.1 What is Machine Learning	2
1.1.2 How Machine Learning works	2
1.1.3 Machine Learning methods	5
1.1.4 History of Machine Learning	6
<b>1.2 VS Code &amp; Dbeaver</b>	<b>8</b>
1.2.1 Introduction to VS Code and Dbeaver	8
1.2.2 Installation and Setup	9
<b>1.3 Tools</b>	<b>11</b>
1.3.1 Numpy	11
1.3.2 Pandas	12
1.3.3 Matplotlib	12

1.3.4 Opencv Python	13
1.3.5 Scikit Learn	13
1.3.6 EasyOcr	14
1.3.7 Pillow	14
1.3.8 Pypng	15
1.3.9 PyQRCode	16
1.3.10 Twilio	16
1.3.11 Mysql Connector Python	17
<b>1.4 Flask</b>	<b>18</b>
1.4.1 Why Flask?	19
1.4.2 Flask used for?	19
1.4.3 Flask Features	19
<b>2. LITERATURE SURVEY</b>	<b>21</b>
2.1 Literature Survey	22
<b>3. FLOW DIAGRAMS</b>	<b>23</b>
3.1 Data Flow Diagram	24
3.2 UML Diagrams	25
<b>4. SYSTEM ANALYSIS</b>	<b>27</b>
4.1 Problem Statement	28
4.2 Existing System	28

4.3 Proposed System	28
4.3.1 Data Collection	29
4.3.2 Model Building	29
4.4 Conclusion	30
4.5 Source Code	31
<b>5. REQUIREMENT AND SPECIFICATION</b>	<b>32</b>
5.1 System Specifications	33
5.1.1 Hardware Requirements	33
5.1.2 Software Requirements	33
<b>6. WEB SCREENS</b>	<b>34</b>
<b>6.1 Model 1</b>	<b>35</b>
6.1.1 Input Page	35
6.1.2 Output Page	36
<b>7. REFERENCES</b>	<b>38</b>
7.1 References	39





## **LIST OF FIGURES**

<b>FIGURES</b>	<b>PAGE NO</b>
1. VS Code Installation & Setup	9
2. DBeaver	10
3. Flask framework architecture	18
4. Data Flow Diagram	24
5. UML Diagrams	25
6. Use Case Diagram	25
7. Sequence Diagram	26
8. Model Building	30
9. Yolo Model	30
10. Input Page	35
11. Output Page	36

# **Chapter 1**

## **INTRODUCTION**

# **1. INTRODUCTION**

## **1.1 Machine Learning**

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

### **1.1.1 What is machine learning?**

Machine learning is a branch of artificial intelligence (AI) focused on building applications that learn from data and improve their accuracy over time without being programmed to do so.

In data science, an algorithm is a sequence of statistical processing steps. In machine learning, algorithms are 'trained' to find patterns and features in massive amounts of data in order to make decisions and predictions based on new data. The better the algorithm, the more accurate the decisions and predictions will become as it processes more data.

Today, examples of machine learning are all around us. Digital assistants search the web and play music in response to our voice commands. Websites recommend products and movies and songs based on what we bought, watched, or listened to before. Robots vacuum our floors while we do . . . something better with our time. Spam detectors stop unwanted emails from reaching our inboxes. Medical image analysis systems help doctors spot tumors they might have missed. And the first self-driving cars are hitting the road.

We can expect more. As big data keeps getting bigger, as computing becomes more powerful and affordable, and as data scientists keep developing more capable algorithms, machine learning will drive greater and greater efficiency in our personal and work lives.

### **1.1.2How Machine Learning Works**

There are four basic steps for building a machine learning application (or model). These are typically performed by data scientists working closely with the business professionals for whom the model is being developed.

### 1.1.3 How Machine Learning Works

There are four basic steps for building a machine learning application (or model). These are typically performed by data scientists working closely with the business professionals for whom the model is being developed.

#### **Step 1: Select and prepare a training data set**

Training data is a data set representative of the data the machine learning model labelled to solve the problem it's designed to solve. In some cases, the training data is labelled data—'tagged' to call out features and classifications the model will need to identify. Other data is unlabeled, and the model will need to extract those features and assign classifications on its own.

In either case, the training data needs to be properly prepared randomized, de-duped and checked for imbalances or biases that could impact the training. It should also be divided into two subsets: the training subset, which will be used to train the application and the evaluation subset, used to test and refine it.

**Step 2: Choose an algorithm to run on the training data set** Again, an algorithm is a set of statistical processing steps. The type of algorithm depends on the type (labelled or unlabeled) and amount of data in the training data set and on the type of problem to be solved.

Common types of machine learning algorithms for use with labelled data include the following:

- **Regression algorithms:** Linear and logistic regressions are examples of regression algorithms used to understand relationships in data. Linear regression is used to predict the value of a dependent variable based on the value of an independent variable. Logistic regression can be used when the dependent variable is binary in nature: A or B. For example, a linear regression algorithm could be trained to predict a salesperson's annual sales (the dependent variable) based on its relationship to the salesperson's education or years of experience (the independent variables.) Another type of regression algorithm called a support vector machine is useful when dependent variables are more difficult to classify.
- **Decision trees:** Decision trees use classified data to make recommendations based on a set of decision rules. For example, a decision tree that recommends betting on a particular horse to win, place, or show could use data about the horse (e.g., age, winning percentage, pedigree) and apply rules to those factors to recommend an action or decision.
- **Instance-based algorithms:** A good example of an instance-based algorithm is K-Nearest Neighbour. It uses classification to estimate how likely a data point is to be a member of one group, or another based on its proximity to other data points.

Algorithms for use with unlabeled data include the following:

- **Clustering algorithms:** Think of clusters as groups. Clustering focuses on identifying groups of similar records and labelling the records according to the group to which they belong. This is done without prior knowledge about the groups and their characteristics. Types of clustering algorithms include the K-means, Two Step, and Kohonen clustering.
- **Association algorithms:** Association algorithms find patterns and relationships in data and identify frequent 'if-then' relationships called association rules. These are like the rules used in data mining.
- **Neural networks:** A neural network is algorithm that defines a layered network of calculations featuring an input layer, where data is ingested; at least one hidden layer, where calculations are performed make different conclusions about input; and an output layer. Where each conclusion is assigned a probability. A deep neural network defines a network with multiple hidden layers, each of which successively refines the results of the previous layer. (For more, see the “Deep learning” section below.)

### **Step 3: Training the algorithm to create the model**

Training the algorithm is an iterative process—it involves running variables through the algorithm, comparing the output with the results it should have produced, adjusting weights and biases within the algorithm that might yield a more accurate result, and running the variables again until the algorithm returns the correct result most of the time. The resulting trained, accurate algorithm is the machine learning model—an important distinction to note, because 'algorithm' and 'model' are incorrectly used interchangeably, even by machine learning mavens.

### **Step 4: Using and improving the model**

The final step is to use the model with new data and, in the best case, for it to improve in accuracy and effectiveness over time. Where the new data comes from will depend on the problem being solved. For example, a machine learning model designed to identify spam will ingest email messages, whereas a machine learning model that drives a robot vacuum cleaner will ingest data resulting from real-world interaction with moved furniture or new objects in the room.

### **1.1.4 Machine Learning methods**

Machine learning methods (also called machine learning styles) fall into three primary categories.

#### **Supervised machine learning**

Supervised machine learning trains itself on a labelled data set. That is, the data is labelled with information that the machine learning model is being built to determine and that may even be classified in ways the model is supposed to classify data. For example, a computer vision model designed to identify purebred German shepherd dogs might be trained on a data set of various labelled dog images.

Supervised machine learning requires less training data than other machine learning methods and makes training easier because the results of the model can be compared to actual labelled results. But properly labelled data is expensive to prepare, and there's the danger of overfitting, or creating a model so closely tied and biased to the training data that it doesn't handle variations in new data accurately.

#### **Unsupervised machine learning**

Unsupervised machine learning ingests unlabeled data—lots and lots of it—and uses algorithms to extract meaningful features needed to label, sort, and classify the data in real-time, without human intervention. Unsupervised learning is less about automating decisions and predictions, and more about identifying patterns and relationships in data that humans would miss. Take spam detection, for example—people generate more email than a team of data scientists could ever hope to label or classify in their lifetimes. An unsupervised learning algorithm can analyze huge volumes of emails and uncover the features and patterns that indicate spam (and keep getting better at flagging spam over time).

#### **Reinforcement Learning**

Reinforcement learning is the third and most advanced algorithm category in Machine Learning. Unlike supervised and unsupervised learning, reinforcement learning continuously improves its model by leveraging feedback from previous iterations. This is different to supervised and unsupervised learning, which both reach an indefinite endpoint after a model is formulated from the training and test data segments. Reinforcement learning can be complicated and is probably best explained through an analogy to a video game. As a player progresses through the virtual space of a game, they learn the value of various actions under different conditions and become more familiar with the field of play. Those learned values then inform and influence a player's subsequent behaviour and their performance immediately improves based on their learning and past experience.

### **1.1.5 History of Machine Learning**

The first case of neural networks was in 1943, when neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper about neurons, and how they work. They decided to create a model of this using an electrical circuit, and therefore the neural network was born.

In 1950, Alan Turing created the world-famous Turing Test. This test is fairly simple - for a computer to pass, it has to be able to convince a human that it is a human and not a computer.

1952 saw the first computer program which could learn as it ran. It was a game which played checkers, created by Arthur Samuel.

Frank Rosenblatt designed the first artificial neural network in 1958, called Perceptron. The main goal of this was pattern and shape recognition.

Another extremely early instance of a neural network came in 1959, when Bernard Widrow and Marcian Hoff created two models of them at Stanford University. The first was called ADELIN, and it could detect binary patterns. For example, in a stream of bits, it could predict what the next one would be. The next generation was called MADELINE, and it could eliminate echo on phone lines, so it had a useful real-world application. It is still in use today.

Despite the success of MADELINE, there was not much progress until the late 1970s for many reasons, mainly the popularity of Von Neumann architecture. This is an architecture where instructions and data are stored in the same memory, which is arguably simpler to understand than a neural network, and so many people-built programs based on this.

Neural networks use back propagation (explained in detail in the Introduction to Neural Networks), and this important step came in 1986, when three researchers from the Stanford psychology department decided to extend an algorithm created by Widrow and Hoff in 1962. This therefore allowed multiple layers to be used in a neural network, creating what are known as 'slow learners', which will learn over a long period of time.

The late 1980s and 1990s did not bring much to the field. However, in 1997, the IBM computer Deep Blue, which was a chess-playing computer, beat the world chess champion. Since then, there have been many more advances in the field, such as in 1998, when research at AT&T Bell Laboratories on digit recognition resulted in good accuracy in detecting handwritten postcodes from the US Postal Service. This used back-propagation, which, as stated above, is explained in detail on the Introduction to Neural Networks.

### **21st Century**

Since the start of the 21st century, many businesses have realized that machine learning will increase calculation potential. This is why they are researching more heavily into it, to stay ahead of the competition.



### Some large projects include:

- **Google Brain (2012)** - This was a deep neural network created by Jeff Dean of Google, which focused on pattern detection in images and videos. It was able to use Google's resources, which made it incomparable to much smaller neural networks. It was later used to detect objects in YouTube videos.
- **Alex Net (2012)** – Alex Net won the Image Net competition by a large margin in 2012, which led to the use of GPUs and Convolution Neural Networks in machine learning. They also created ReLU, which is an activation function that greatly improves efficiency of CNNs.
- **Deep Face (2014)** - This is a Deep Neural Network created by Facebook, which they claimed can recognize people with the same precision as a human can.
- **Deep Mind (2014)** - This company was bought by Google and can play basic video games to the same levels as humans. In 2016, it managed to beat a professional at the game Go, which is considered to be one the world's most difficult board games.
- **Open AI (2015)** - This is a non-profit organization created by Elon Musk and others, to create safe artificial intelligence that can benefit humanity.
- **Amazon Machine Learning Platform (2015)** - This is part of Amazon Web Services and shows how most big companies want to get involved in machine learning. They say it drives many of their internal systems, from regularly used services such as search recommendations and Alexa, to more experimental ones like Prime Air and Amazon Go.
- **ResNet (2015)** - This was a major advancement in CNNs, and more information can be found on the Introduction to CNNs page.
  - **U-net (2015)** - This is a CNN architecture specialized in biomedical image segmentation. It introduced an equal amount of up sampling and down sampling layers, and also skip connections. More information on what this means can be found on the Semantic Segmentation page.

### 1.1.6 Introduction to Python

Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science. Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to

## 1.2.1 Introduction to VS Code and Dbeaver

### Introduction to VS Code

Visual Studio Code, also known as VS Code, is a free and open-source source code editor developed by Microsoft for Windows, Linux, and macOS. It is designed to be highly customizable and flexible, while also providing a user-friendly interface for editing, debugging, and managing code.

Some key features of VS Code include:

- Support for a wide range of programming languages, including JavaScript, Python, C++, and more.
- A powerful debugger for diagnosing and fixing errors in your code.
- Built-in Git integration for version control and collaboration with others.
- An extensive library of extensions and plugins to customize and extend the functionality of the editor.
- Cross-platform compatibility, allowing you to work seamlessly across different operating systems.

Overall, VS Code is a popular choice among developers due to its ease of use, flexibility, and powerful set of features. Whether you are a beginner or an experienced developer, VS Code provides a solid foundation for managing and editing your code.

### Introduction to DBeaver: A Universal Database Tool for Developers and Analysts"

DBeaver is a free and open-source universal database tool for developers, database administrators, and analysts. It provides a graphical user interface for managing and querying various databases, including MySQL, PostgreSQL, Oracle, and Microsoft SQL Server, among others.

Some of the key features of DBeaver include:

- Support for a wide range of databases, including relational databases, NoSQL databases, and cloud-based databases.
- A user-friendly interface for creating, editing, and executing SQL queries and scripts.
- Advanced data visualization and editing capabilities.
- Built-in data modeling tools for designing and manipulating database schemas.
- Support for database administration tasks such as backup, restore, and security management.
- Extensive plugin architecture for extending the functionality of the tool.

Overall, DBeaver is a powerful and versatile tool for working with databases, offering a wide range of features and support for many different database systems. It is a great choice for anyone who needs to work with databases on a regular basis, from beginners to experienced professionals.

## 1.2.2 Installation and Setup

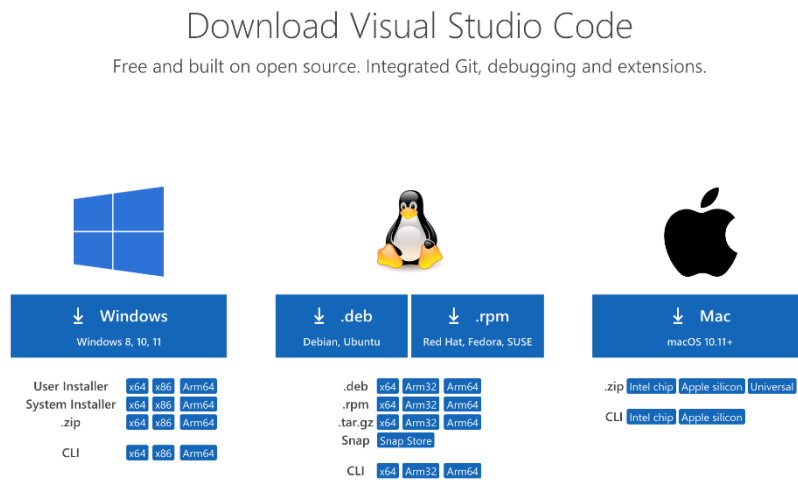


Fig.1

### VS Code Installation and Setup

1. Go to the official website of Visual Studio Code at <https://code.visualstudio.com/download>
2. Click on the "Download for [your operating system]" button.
3. Once the download is complete, open the installer file.
4. Follow the prompts in the installer to complete the installation process.
5. Once the installation is complete, launch VS Code.

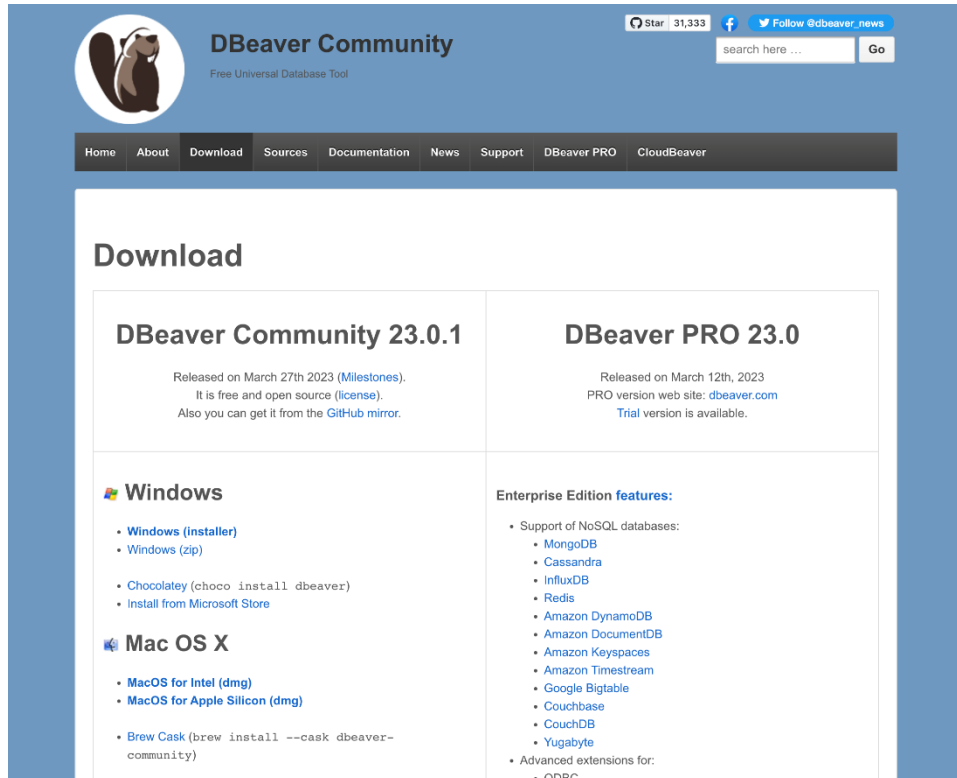


Fig.2

1. Go to the official website of DBeaver at <https://dbeaver.io/>
2. Click on the "Download" button on the homepage.
3. Select your operating system from the list of options.
4. Choose the appropriate installer package for your system architecture (32-bit or 64-bit).
5. Once the download is complete, open the installer file.
6. Follow the prompts in the installer to complete the installation process.
7. Once the installation is complete, launch DBeaver.

## 1.3 Tools

### 1.3.1 Numpy

NumPy (short for Numerical Python) is a Python library that provides support for large, multi-dimensional arrays and matrices, as well as a variety of mathematical functions for working with this data. NumPy is a fundamental library for scientific computing with Python, and it is widely used in the fields of data analysis, machine learning, and scientific research.

NumPy is built around the concept of ndarrays, which are N-dimensional arrays of homogeneous data types. The ndarrays can have any number of dimensions and can contain various data types, including integers, floating-point numbers, and complex numbers. NumPy also provides a set of functions for performing mathematical operations on these arrays, such as addition, subtraction, multiplication, division, and more.

In addition to its support for ndarrays, NumPy also provides a number of other features for scientific computing and data analysis. These include:

- Linear algebra: NumPy provides a set of functions for performing linear algebra operations such as matrix multiplication, inversion, and decomposition.
- Fourier analysis: NumPy provides functions for performing Fourier transforms and other signal processing operations.
- Random number generation: NumPy includes a powerful set of functions for generating random numbers, which are useful for simulations and other applications.
- Tools for integrating with other data analysis libraries: NumPy integrates seamlessly with other Python libraries such as Pandas and Matplotlib, making it easy to perform complex data analysis tasks.

Overall, NumPy is an essential library for scientific computing and data analysis with Python, and it is widely used in many different fields, including physics, engineering, finance, and more.

To use NumPy in your Python project, you need to install the NumPy library first. Here's how to do it:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install NumPy using pip, the package installer for Python: *pip install numpy*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful:
5. Successfully installed numpy-1.21.4
6. Once NumPy is installed, you can start using it in your Python code. To import NumPy, simply add the following line to the beginning of your Python script: *import numpy as np*
7. This line imports NumPy and creates an alias for it, so you can refer to it as "np" in your code.

### 1.3.2 Pandas

Pandas is a popular Python library used for data manipulation and analysis. It provides powerful data structures for handling and analyzing data, as well as a variety of tools for data cleaning, reshaping, and visualization.

Some key features of Pandas include:

- Data structures for working with both labeled and unlabeled data, including Series (1D) and DataFrame (2D).
- Efficient data manipulation tools for merging, grouping, filtering, and transforming data.
- Built-in support for handling missing or incomplete data.
- A variety of data visualization tools for creating charts, graphs, and other visual representations of data.
- Integration with other Python libraries for scientific computing, such as NumPy and Matplotlib.

To install Pandas, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install Pandas using pip, the package installer for Python: *pip install pandas*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful:
5. Successfully installed pandas-1.3.4
6. Once Pandas is installed, you can start using it in your Python code. To import Pandas, simply add the following line to the beginning of your Python script: *import pandas as pd*
7. This line imports Pandas and creates an alias for it, so you can refer to it as "pd" in your code.

### 1.3.3 Matplotlib

Matplotlib is a popular Python library used for creating high-quality 2D and 3D plots and charts. It provides a variety of tools for data visualization, including line plots, scatter plots, bar charts, histograms, and more.

Some key features of Matplotlib include:

- Support for a wide range of plot types, including 2D and 3D plots, scatter plots, histograms, and more.
- Highly customizable plots, with options for changing colors, fonts, labels, and other visual elements.
- Integration with other Python libraries for scientific computing, such as NumPy and Pandas.
- Support for interactive plots, allowing users to zoom, pan, and modify plots on the fly.

To install Matplotlib, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install Matplotlib using pip, the package installer for Python: *pip install matplotlib*
3. Note: Make sure you have Python and pip installed on your system before running this command.

4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed matplotlib-3.4.3
5. Once Matplotlib is installed, you can start using it in your Python code. To import Matplotlib, simply add the following line to the beginning of your Python script: *import matplotlib.pyplot as plt*
6. This line imports the Pyplot module of Matplotlib and creates an alias for it, so you can refer to it as "plt" in your code.

### 1.3.4 OpenCv Python

OpenCV (OpenSource Computer Vision Library) is an open-source computer vision and machine learning software library. It has a wide range of applications, from real-time computer vision and facial recognition to image and video processing.

To install OpenCV for Python, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install OpenCV using pip, the package installer for Python: *pip install opencv-python*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful:
5. Successfully installed opencv-python-4.5.4.58
6. Once OpenCV is installed, you can start using it in your Python code. To import OpenCV, simply add the following line to the beginning of your Python script: *import cv2*
7. This line imports OpenCV and makes it available for use in your code.

### 1.3.5 Scikit Learn

Scikit-learn (also known as sklearn) is a popular Python library for machine learning. It provides a wide range of tools and algorithms for data analysis, modeling, and prediction.

Some key features of Scikit-learn include:

- A variety of supervised and unsupervised learning algorithms, including classification, regression, clustering, and dimensionality reduction.
- Integration with other Python libraries for scientific computing, such as NumPy and Pandas.
- Built-in support for data preprocessing, including feature scaling and normalization.
- A wide range of evaluation metrics for assessing model performance, including accuracy, precision, recall, and F1 score.
- Support for model selection and hyperparameter tuning, including cross-validation and grid search.

To install Scikit-learn, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install Scikit-learn using pip, the package installer for Python: *pip install scikit-learn*

3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed scikit-learn-1.0
5. Once Scikit-learn is installed, you can start using it in your Python code. To import Scikit-learn, simply add the following line to the beginning of your Python script: *import sklearn*
6. This line imports Scikit-learn and makes it available for use in your code.

### 1.3.6 Easyocr

EasyOCR is a Python library for optical character recognition (OCR). It provides a simple and easy-to-use interface for extracting text from images and other types of media.

Some key features of EasyOCR include:

- Support for over 70 languages, including English, Chinese, Japanese, Korean, and many others.
- Integration with other Python libraries for image processing, such as OpenCV and Pillow.
- Built-in support for multiple OCR engines, including Tesseract and CuneiForm.
- Easy-to-use API for extracting text from images and other types of media.
- Support for both CPU and GPU processing.

To install EasyOCR, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install EasyOCR using pip, the package installer for Python: *pip install easyocr*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed easyocr-1.3.2
5. Once EasyOCR is installed, you can start using it in your Python code. To import EasyOCR, simply add the following line to the beginning of your Python script: *import easyocr*
6. This line imports EasyOCR and makes it available for use in your code.

### 1.3.7 Pillow

Pillow is a popular Python library for image processing and manipulation. It provides a wide range of tools and functions for working with images, including opening and saving images in various formats, resizing and cropping images, applying filters and effects, and much more.

Some key features of Pillow include:

- Support for a wide range of image file formats, including JPEG, PNG, BMP, and many others.
- A simple and intuitive API for working with images, including loading and saving images, and applying various transformations and filters.
- Support for advanced image processing tasks, such as image enhancement, segmentation, and object detection.
- Integration with other Python libraries for scientific computing, such as NumPy and Scikit-learn.
- Support for both CPU and GPU processing.



To install Pillow, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install Pillow using pip, the package installer for Python: *pip install pillow*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed pillow-8.3.2
5. Once Pillow is installed, you can start using it in your Python code. To import Pillow, simply add the following line to the beginning of your Python script: *from PIL import Image*
6. This line imports the Image module from Pillow and makes it available for use in your code.

### 1.3.8 Pypng

PyPNG is a Python library for reading and writing PNG (Portable Network Graphics) image files. It provides a simple and easy-to-use interface for working with PNG images in Python.

Some key features of PyPNG include:

- Support for reading and writing PNG images with various bit depths, color types, and compression levels.
- Integration with other Python libraries for image processing, such as Pillow and NumPy.
- Support for both pure Python and C extensions, depending on your needs and performance requirements.

To install PyPNG, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install PyPNG using pip, the package installer for Python: *pip install pypng*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed pypng-0.0.21
5. Once PyPNG is installed, you can start using it in your Python code. To import PyPNG, simply add the following line to the beginning of your Python script: *import png*
6. This line imports the PyPNG module and makes it available for use in your code.

### 1.3.9 PyQRCode

PyQRCode is a Python library for generating QR (Quick Response) codes. It provides a simple and easy-to-use interface for creating QR codes in Python.

Some key features of PyQRCode include:

- Support for creating QR codes of various sizes and error correction levels.
- Integration with other Python libraries for image processing, such as Pillow and NumPy.
- Support for generating QR codes with custom logos or images embedded.

To install PyQRCode, you can follow these steps:

1. Open your command prompt or terminal.
2. Type in the following command and press enter to install PyQRCode using pip, the package installer for Python: *pip install pyqrcode*
3. Note: Make sure you have Python and pip installed on your system before running this command.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed pyqrcode-1.2.1
5. Once PyQRCode is installed, you can start using it in your Python code. To import PyQRCode, simply add the following line to the beginning of your Python script: *import pyqrcode*
6. This line imports the PyQRCode module and makes it available for use in your code.

### 1.3.10 Twilio

Twilio is a cloud communications platform that allows developers to programmatically send and receive text messages, voice calls, and other forms of communication. It provides a simple and easy-to-use API for integrating communication capabilities into your software applications.

To use Twilio in your Python projects, you need to follow these steps:

1. Sign up for a Twilio account at <https://www.twilio.com/try-twilio>. You will need to provide some basic information and verify your email address.
2. Once you have signed up, you will be given an Account SID and an Auth Token. These are required to authenticate your requests to the Twilio API.
3. Install the Twilio Python helper library using pip, the package installer for Python. Open your command prompt or terminal and type the following command:
4. Copy code
5. *pip install twilio*
6. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed twilio-6.63.0.
7. Once the Twilio helper library is installed, you can start using it in your Python code. To send a text message, for example, you can use the following code:

```

from twilio.rest import Client

# Your Account SID and Auth Token from twilio.com/console
account_sid = 'your_account_sid'
auth_token = 'your_auth_token'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
        body="Hello from Twilio!",
        from_='+1234567890', # Replace with your Twilio number
        to='+9876543210' # Replace with your mobile number
    )

print(message.sid)

```

8. This code sends a text message to the specified mobile number using your Twilio account. You will need to replace the **your\_account\_sid** and **your\_auth\_token** variables with your actual Account SID and Auth Token, which you can find on the Twilio console.

On the other hand, if NumPy and Scipy is not yet installed on your Python workstation then, you can install them by using either pip or conda.

Another option to use scikit-learn is to use Python distributions like Canopy and Anaconda because they both ship the latest version of scikit-learn.

### 1.3.11 Mysql connector python

MySQL Connector/Python is a Python library that provides a standardized database driver for connecting Python applications to MySQL databases. It is a pure Python implementation of the MySQL client/server protocol and is fully compliant with the Python Database API specification.

To use MySQL Connector/Python in your Python projects, you need to follow these steps:

1. Install MySQL Connector/Python using pip, the package installer for Python. Open your command prompt or terminal and type the following command:
2. Copy code
3. pip install mysql-connector-python.
4. Wait for the installation to complete. You should see output similar to the following if the installation is successful: Successfully installed mysql-connector-python-8.0.27

5. Once MySQL Connector/Python is installed, you can start using it in your Python code. To connect to a MySQL database, for example, you can use the following code:

```
import mysql.connector
# Establish a connection to the database
cnx = mysql.connector.connect(user='your_username', password='your_password',
                              host='your_hostname', database='your_database')

# Perform a query
cursor = cnx.cursor()
query = ("SELECT * FROM your_table")
cursor.execute(query)

# Process the results
for (column1, column2, column3) in cursor:
    print("{}\t{}\t{}".format(column1, column2, column3))

# Clean up
cursor.close()
cnx.close()
```

6. This code connects to a MySQL database using the specified username, password, hostname, and database name. It then performs a SELECT query on a table and processes the results. You will need to replace the **your\_username**, **your\_password**, **your\_hostname**, **your\_database**, and **your\_table** variables with your actual database credentials and table name.

## 1.4 Flask

Flask is a popular micro web framework for building web applications in Python. It is lightweight and easy to use, making it a popular choice for beginners and experienced developers alike. Flask is designed to be simple and flexible, providing a basic set of features that can be extended as needed using third-party extensions.

One of the key features of Flask is its use of routes, which allow developers to define URLs and associate them with functions that handle requests to those URLs. Flask also provides built-in support for handling forms, cookies, and sessions, as well as integrating with various databases and authentication systems.

Overall, Flask is a great choice for building small to medium-sized web applications that require flexibility and customization. Its simplicity and ease of use make it a popular choice among Python developers, and its active community of contributors provides a wide range of extensions and resources for developers to use.



### 1.4.1 Why Flask?

Now, you would be able to understand, when we define Flask as a micro-framework that is built using WSGI and Jinja 2 template engine.

Major advantages of Flask are:

1. Ease of setup and use.
2. Freedom to build the structure of the web application.

With freedom comes responsibility, similarly, Flask needs the developers to carefully structure it, since Flask doesn't have "flask rules" to follow as compared to frameworks like Django. As the web app increases in complexity, this structuring is what is going to be the foundation.

### 1.4.2 Flask used for?

Flask is used for building web applications and APIs using Python. It is a micro web framework that provides a basic set of features and tools for developing web applications, while also allowing developers to extend and customize its functionality using third-party packages.

Here are some common use cases for Flask:

1. Building small to medium-sized web applications: Flask's simplicity and flexibility make it a great choice for building small to medium-sized web applications, such as blogs, online stores, and social media platforms.
2. Developing APIs: Flask can also be used for building APIs (Application Programming Interfaces) that allow other applications or services to access and interact with your web application's data and functionality.
3. Rapid prototyping: Flask's lightweight nature and ease of use make it an ideal choice for rapidly prototyping web applications or testing out new ideas.
4. Customizing larger web applications: Flask can be used to customize and extend the functionality of larger web applications, such as content management systems (CMS) or e-commerce platforms, by creating custom plugins or extensions.

Overall, Flask is a versatile web framework that can be used for a variety of web development tasks, from building small web applications to developing complex APIs and customizing larger applications.

### 1.4.3 Flask Features

1. Lightweight and flexible: Flask is a lightweight web framework that provides only the essential features required for building web applications. It is also very flexible and can be easily customized and extended using third-party packages.
2. Routes and URL mapping: Flask allows developers to define routes and map URLs to specific functions, making it easy to handle incoming requests and build web pages dynamically.
3. Built-in development server: Flask comes with a built-in development server, making it easy to test and debug web applications without the need for additional server software.
4. Templating engine: Flask includes a templating engine that allows developers to create dynamic HTML templates using variables, loops, and conditional statements.

5. Support for various data formats: Flask supports a wide range of data formats, including JSON, XML, and YAML, making it easy to build web APIs that can communicate with other applications.
6. Integration with popular databases: Flask integrates with a variety of databases, including SQLite, MySQL, and PostgreSQL, making it easy to store and retrieve data in web applications.

## **Chapter 2**

# **LITERATURE SURVEY**

## **2. Literature Survey**

### **2.1 Literature Survey**

The proposed smart parking system, SmartParkX, is an innovative solution that aims to automate the parking process and provide a convenient parking experience to customers. Several technologies are incorporated into the system to achieve this goal.

The use of Automatic Number Plate Recognition (ANPR) technology during the Entry and Exit phases eliminates the need for manual ticketing or verification, which streamlines the process and reduces waiting times. ANPR has been widely used in various applications, including parking management systems, toll collection systems, and traffic monitoring systems. Studies have shown that ANPR can accurately and quickly read license plates, making it an efficient technology for parking management systems (Fang et al., 2020; Ma et al., 2019).

The system also employs object detection algorithms such as YOLO to identify the type of vehicle as a 2-wheeler or 4-wheeler. Object detection algorithms have been used in various applications, including autonomous vehicles, robotics, and surveillance systems. YOLO, in particular, is known for its high accuracy and real-time performance (Redmon et al., 2016). By using YOLO, SmartParkX can allocate the appropriate parking slot and calculate the parking fee accurately based on the type of vehicle.

The Space Counter algorithm is used to allocate an available parking slot based on the type of vehicle. This algorithm helps to manage the parking lot efficiently and ensures that the available slots are allocated to the vehicles correctly based on their type. Various parking management systems have employed different algorithms to allocate parking spaces, such as genetic algorithms (Cheng et al., 2019) and fuzzy logic (Gong et al., 2017). The Space Counter algorithm used in SmartParkX is an effective and efficient solution for parking allocation.

In conclusion, SmartParkX is a highly innovative and sophisticated parking management system that incorporates various technologies to provide a convenient and efficient parking experience to customers. The system leverages ANPR, object detection algorithms like YOLO, and the Space Counter algorithm to automate the parking process and allocate parking spaces accurately. The literature review indicates that these technologies have been widely used and have shown promising results in various applications, making SmartParkX a promising solution for parking management.



## **Chapter 3**

# **FLOW DIAGRAMS**

### 3. Flow Diagrams

#### 3.1 Data Flow Diagram

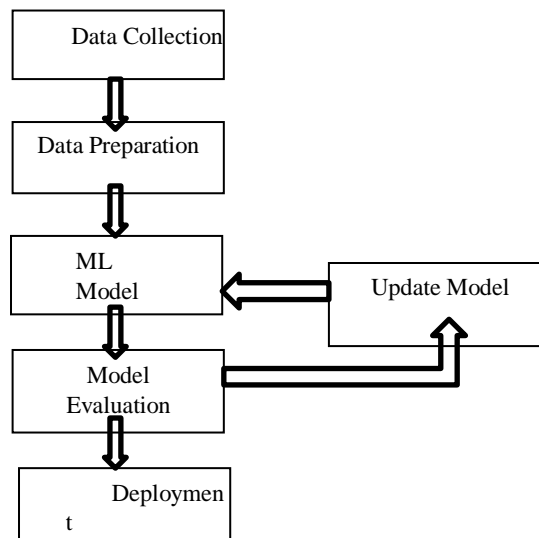


Fig.11

#### Data Collection:

Data Collection involves collecting the data as per the requirements of the project. In this paper we have performed two experiments with two different data-sets respectively. The first data-set was obtained from kaggle. The data set consists of 300 records.

#### Data Preparation:

Data preparation involves the cleaning and exploration of data to find relationships among the features of the data. In first experiment the data was inconsistent and had duplicates. These had to be removed and the data-set is cleaned. The second data-set had many features and the data is explored and the right features were selected for the utilization of the model.

#### ML Model Training:

Initializing the ML models and fitting them to train. We have used various classifiers like Logistic Regression, SVM, KNN, Decision Tree, Random Forest and Gradient Boosting Classifier. The training to testing ratio for the first data-set is 80/20 and 70/30 for the second data-set.

#### Model Evaluation:

This phase involves evaluating the models to see the performance of each model. Both experiments were evaluated by using confusion matrix and accuracy score.

### Update Model:

Update the model parameters to improve the performance. We have performed hyper-parameter tuning to each and every algorithm to find the best parameters.

### Deployment:

Deployment is to host the application in some cloud platform. We have deployed our models in Heruko cloud platform.

## 3.1 UML Diagrams

### 3.1.1 Use Case Diagram

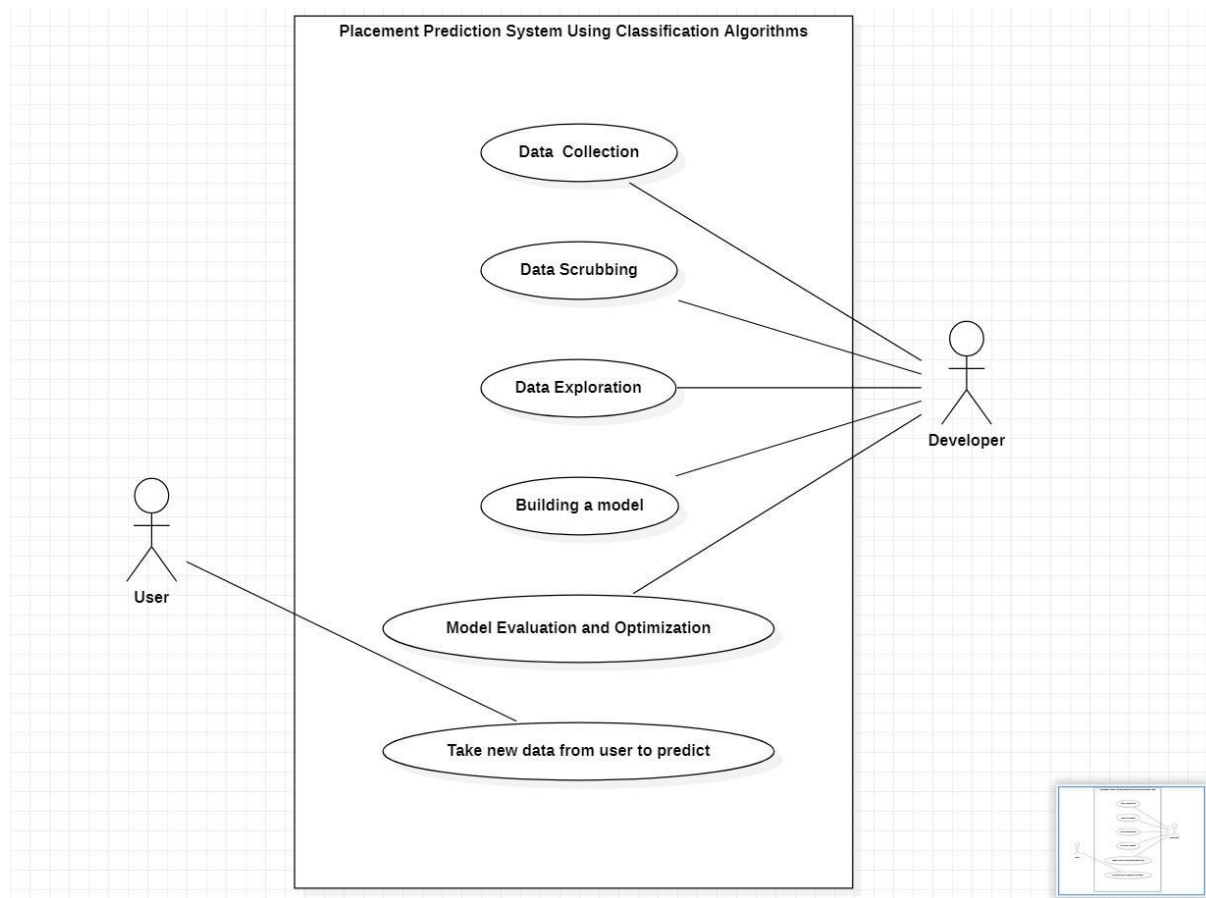


Fig.12

This is a use case diagram which consists of different phases involved in this project. It consists of different parts which are – Use cases and actors. The actors are user and developer who can interact with the use cases. The developer interacts with different phases such as – ‘Data collection’, ‘Data Scrubbing’, ‘Data Exploration’, ‘Building a model’, ‘Model Evaluation and Optimization’. The user interacts with a single use case – ‘Take new data from user to pre

### 3.1.2 Sequence Diagram

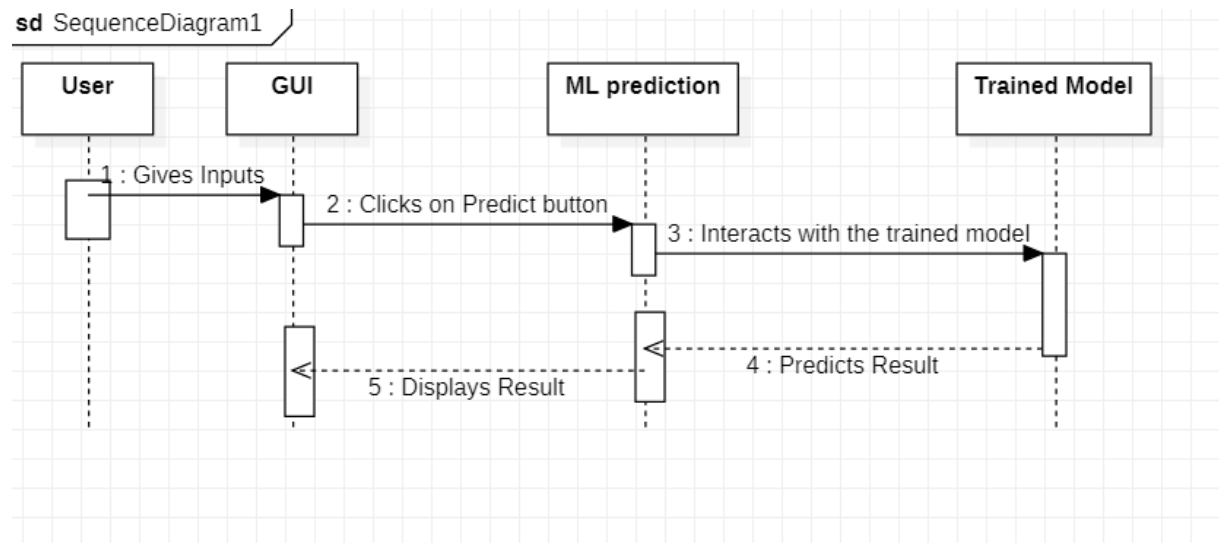


Fig.13

This is a sequence diagram and it contains the sequence of the flow of execution.

The user sends the data to the model through the UI and the pre-trained predicts the outcome and send the result response back to the UI where the user can view the results

## **Chapter 4**

# **SYSTEM ANALYSIS**

## **4. SYSTEM ANALYSIS**

### **4.1 Problem Statement**

SmartParkX system is focused on addressing the inconvenience and inefficiency of traditional parking systems. Traditional parking systems often rely on manual ticketing or verification, which can lead to long wait times and frustrating experiences for customers. In addition, the allocation of parking spaces is often done manually, which can lead to errors and inefficient use of space. The SmartParkX system aims to solve these problems by leveraging advanced technologies such as Automatic Number Plate Recognition and object detection to automate the parking process and provide a convenient and efficient parking experience for customers. By doing so, the system can eliminate the need for manual ticketing and verification, ensure accurate allocation of parking spaces, and provide real-time information on parking availability and fees.

### **4.2 Existing System**

The existing traditional parking systems rely on manual ticketing or verification processes, which can result in long wait times and a frustrating experience for customers. Customers are required to take a ticket upon entering the parking lot and then pay the fee upon exiting, often leading to confusion and inconvenience. The allocation of parking spaces is also done manually, leading to errors and inefficient use of space. Moreover, there is no real-time information available on parking availability, leading to uncertainty and frustration for customers. The traditional parking systems are therefore inefficient and do not provide a satisfactory experience for customers.

### **4.3 Proposed System**

The proposed SmartParkX system leverages advanced technologies to automate the parking process and provide a convenient and efficient parking experience to customers. The system employs ANPR technology to read the vehicle's license plate as it approaches the parking lot, eliminating the need for manual ticketing or verification and ensuring a smooth and hassle-free entry process. The YOLO object detection algorithm is used to identify the type of vehicle and allocate the appropriate parking slot, while the Space Counter algorithm efficiently manages the parking lot and ensures that the available slots are allocated correctly based on the type of vehicle. The system calculates the parking fee in real-time based on fixed rates for 2-wheelers and 4-wheelers and displays it on a QR code. Upon successful payment, the system updates the check-out time and frees up the parking slot for the next vehicle. The SmartParkX system provides a superior parking experience to customers and ensures efficient use of parking spaces.

### 4.3.1 Model 1

The data collection process for the parking lot management system can be described as follows:

The system first gathers the following information from the user:

Vehicle number

Vehicle type

Phone number

This information is then stored into a data set. Once the data set is created, the following steps are performed:

The ANPR function is used to fetch the number plate of the vehicle.

The Space Counter function is called to determine the available parking slots.

The Object Size function is used to determine whether the vehicle is a two-wheeler or a four-wheeler.

The Timing function and the Date function are called to record the check-in time and check-in date of the vehicle.

After the above steps are completed, the received data is pushed into a Cloud DB (MySQL) through the following queries:

Vehicle number

Vehicle type

Slot allocated

Phone number

Check-in time

Check-out time

Finally, data cleaning operations are performed on the dataset to remove any noise and normalize the features to a single scale to improve the performance of the system.

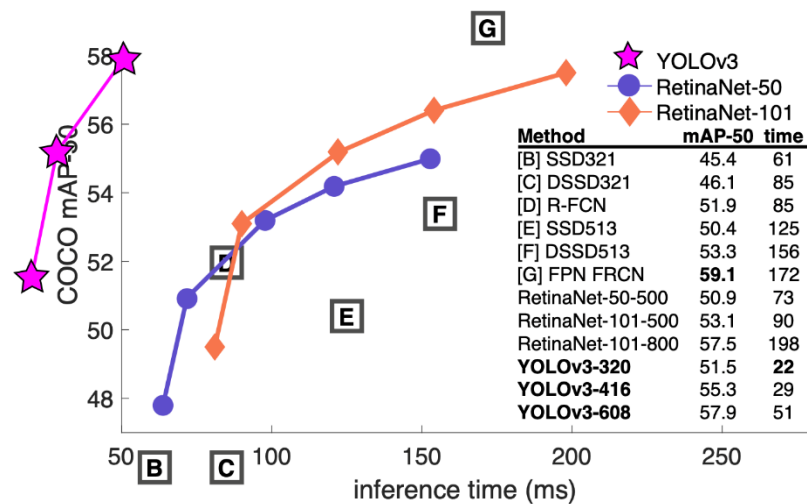
### 4.3.2 Model 2

#### Model Building

In the parking lot management system, the collected data of the users' vehicle number, vehicle type, and phone number are used to allocate parking slots and record the check-in time and date. The system utilizes functions such as the Space Counter, Timing, and Date to provide real-time information to the customer about available parking slots and the check-in process.

Shifting our focus to YOLOv3, an object detection system, it shows good performance in terms of the AP50 metric, which measures the mean average precision at an IOU threshold of 0.5. However, its accuracy decreases as the IOU threshold increases, indicating challenges in box-object alignment. With the implementation of new multi-scale predictions, YOLOv3 has improved its performance on small objects. However, its accuracy still lags behind on medium and larger objects. Nevertheless, YOLOv3 outperforms other detection systems in terms of speed and accuracy.

	backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	<b>52.1</b>
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	<b>40.8</b>	<b>61.1</b>	<b>44.1</b>	<b>24.1</b>	<b>44.2</b>	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9



## 4.4 Conclusion

In conclusion, the proposed SmartParkX system offers a solution to the inefficiencies and frustrations associated with traditional parking systems. By leveraging advanced technologies such as ANPR and object detection algorithms, the proposed system automates the parking process and provides a seamless and efficient experience to customers. The use of real-time parking availability information and automated payment processes further enhances the customer experience. Data collection plays a crucial role in the development of the system, ensuring accurate recognition of license plates, vehicle types, and efficient allocation of parking spaces. The SmartParkX system has the potential to revolutionize the parking industry, providing a more efficient and convenient parking experience to customers while ensuring optimal use of parking spaces.



## **4.5 Source code**

### **4.5.1 Model 1**

Link to source code:

<https://mega.nz/file/WmYngCZJ#R0h4oNXnAduJP9DJzRwUU7d7hMulFzbkR3kE8D3LQRs>

## **Chapter 5**

# **REQUIREMENT AND SPECIFICATION**

## **5.REQUIREMENT AND SPECIFICATION**

### **5.1 System Specifications**

#### **5.1.1 Hardware Requirements**

- Processor : multi core processor
- RAM : 8GB (minimum)
- Hard Disk : 3GB (minimum)

#### **5.1.2 Software requirements**

- Software's To Be Installed : VS Code, Dbeaver
- Programming Language : Python
- Framework : Flask
- Libraries : Numpy, Pandas, Matplotlib, Scikit learn, Easyocr, Pillow, Pypng, PyQRCode, Twilio, Mysql connector python

## **Chapter 6**

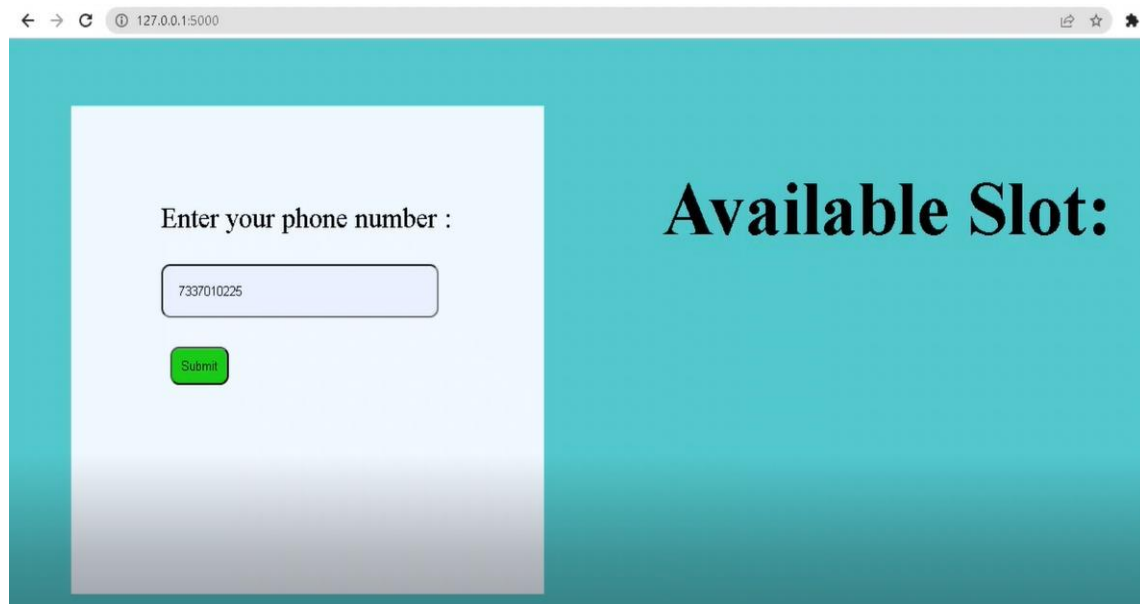
# **APP SCREENS**

## 6.Web page Screens

### 6.1 Model 1

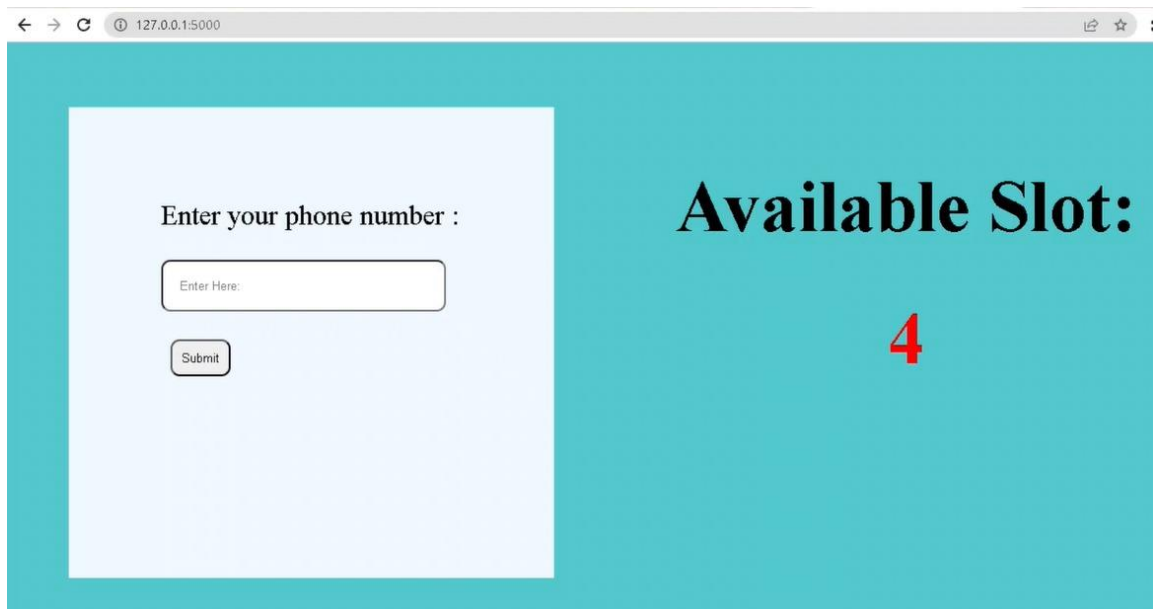
#### Sample Website

##### 6.1.1 Input Page:



A screenshot of a web browser displaying a form on the left and a large teal area on the right. The form has a label "Enter your phone number :", a text input field containing "7337010225", and a green "Submit" button. The teal area on the right contains the text "Available Slot:" in a large, bold, black font.

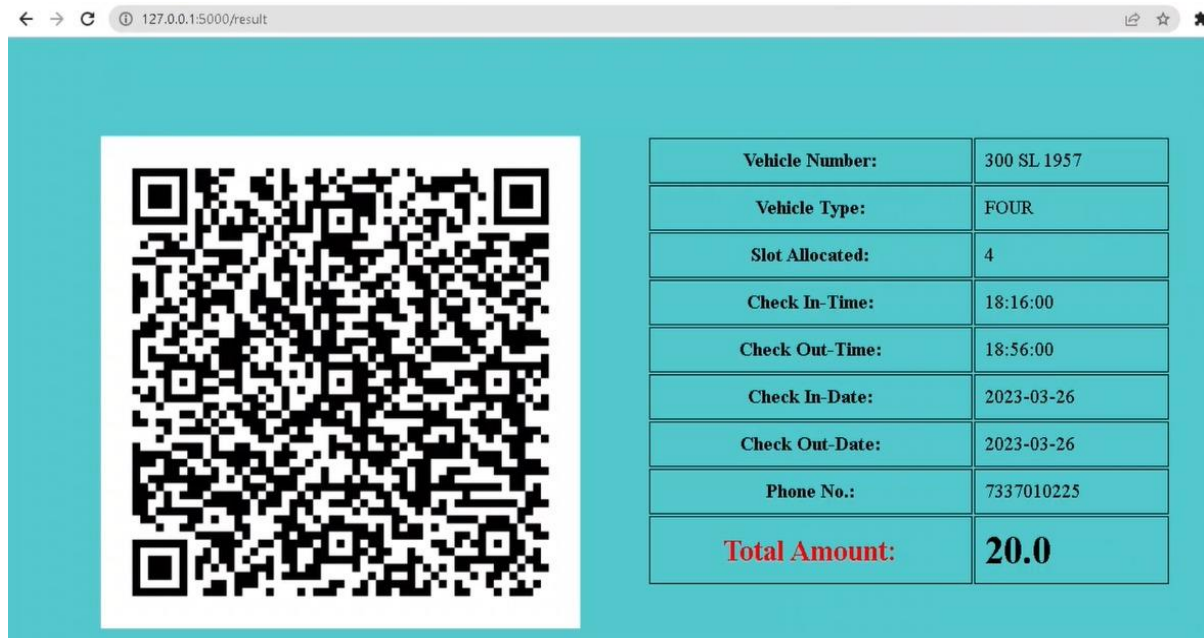
The customer enters their mobile number on the input page of the system.



A screenshot of a web browser displaying a form on the left and a large teal area on the right. The form has a label "Enter your phone number :", a text input field containing "Enter Here:", and a "Submit" button. The teal area on the right contains the text "Available Slot:" in a large, bold, black font, and a large red number "4" below it.

The system checks the parking lot availability for the customer's vehicle type. If there are available parking slots for the customer's vehicle type, the system assigns a slot to the customer and displays the slot number on the screen.

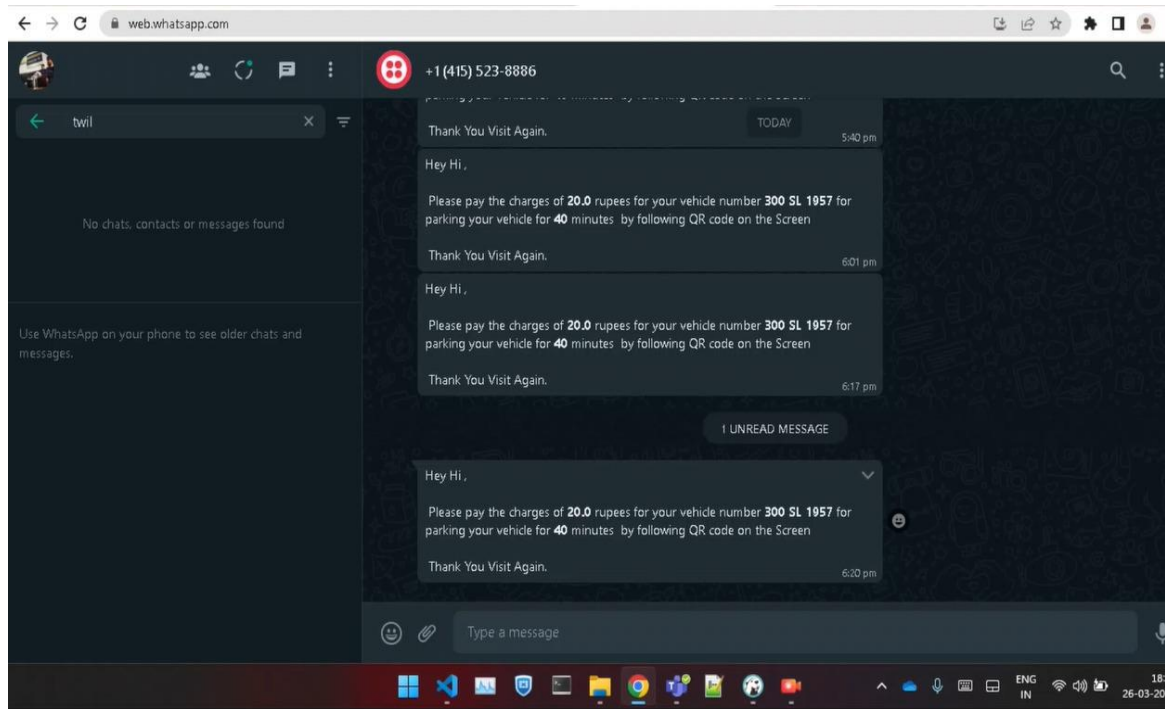
### 6.1.2 Output Page:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/result". The main content area has a light blue background. On the left, there is a large QR code. On the right, there is a table with the following data:

Vehicle Number:	300 SL 1957
Vehicle Type:	FOUR
Slot Allocated:	4
Check In-Time:	18:16:00
Check Out-Time:	18:56:00
Check In-Date:	2023-03-26
Check Out-Date:	2023-03-26
Phone No.:	7337010225
<b>Total Amount:</b>	<b>20.0</b>

1. As the vehicle approaches the exit gate, the ANPR module of the SmartParkX system reads the license plate and checks if the vehicle is in the database.
2. If the vehicle is in the database, the system retrieves the check-in date and time.
3. The system calculates the duration of the parking stay based on the check-in date and time and the current exit time.
4. Based on the vehicle type and the duration of the parking stay, the system calculates the parking fee using a fixed rate.
5. The parking fee is displayed on a QR code, which the customer can scan and pay using their preferred payment method.
6. The system updates the check-out date and time in the database and frees up the parking slot for the next vehicle.



1. The SmartParkX system send a reminder message to the customer's registered whatsapp mobile number to pay the parking fee.
2. The reminder message contains details such as the vehicle number and the total amount to be paid.
3. The customer can then scan the QR code and complete the payment process.

## **Chapter 7**

# **REFERENCES**



## 7. REFERENCES

### 7.1 References

[1] S.Suraj, N.K.Sridhar, J.J.Jijesh, Shivashankar. "Automatic Parking Gateway using Character Recognition", 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology(RTEICT), 2018. [Publication].

[2] Kartikeya Jain, Tanupriya Choudhury, Nirbhay Kashyap, "Smart vehicle identification system using OCR", 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), 2017, [Publication].

[3] Automatic Number Plate Recognition System (ANPR): A Survey. [Online]. Available: [https://www.researchgate.net/publication/236888959\\_Automatic\\_Number\\_Plate\\_Recognition\\_System\\_ANPR\\_A\\_Survey](https://www.researchgate.net/publication/236888959_Automatic_Number_Plate_Recognition_System_ANPR_A_Survey)

[4] Open Data Science, 'Overview of the YOLO Object Detection Algorithm'. [Online]. Available: <https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0yoloyolov-2-28b1b93e2088>