**Internship report**

### A report Submitted in partial fulfilment of the requirements

### For the Industry Internship programs

### Submitted by

### Vinay M 1SG21EE082

### Shravani S 1SG21EE065

### Tejashree K 1SG21EE075

### Under the supervision of

### DR Rekha SN



### Sapthagiri college of engineering

### Department of Electrical and Electronics Engineering

### Academic year

### 2020-21

**Vehicle Movement Analysis & Insight Generation in College Campus using edge AI**

# CHAPTER 1

## INTRODUCTION :

The primary objective of this project is to develop an Edge AI-based solution that can analyse vehicle Movement in and out of a college campus using data from cameras capturing vehicle photos and license plates. The solution provides insights on vehicle movement patterns, parking occupancy, and match vehicles to an approved vehicle database.

This project processes parking lot data from an Excel file to analyze and visualize slot usage and occupancy over time. The project determines how each slot is used over time. Then generates visualizations to help understand the usage patterns and peak times. Finally, saves the processed data and key metrics for further analysis.

## DATA DESCRIPTION :

The dataset used for the analysis consists of vehicle entry and exit times in a parking lot. Here are the key components:

Columns

Vehicle ID:

Description: A unique identifier for each vehicle.

Type: String/Integer.

Entry Time:

Description: The time when the vehicle entered the parking lot.

Type: Time (formatted as HH:MM:SS).

Exit Time:

Description: The time when the vehicle left the parking lot.

Type: Time (formatted as HH:MM:SS).

Purpose : The dataset is used to analyze the usage and occupancy of parking slots over time, identify peak usage periods, and provide insights for better parking lot management

## METHODOLOGY

1.Data Processing:

* Pandas: For data manipulation, sorting, and creating time indices.
* Datetime: To convert entry and exit times into a datetime format for accurate calculations.

2.Data Visualization:

* Matplotlib: For plotting graphs such as bar plots and line charts to visualize slot usage and occupancy over time.
* Seaborn: For creating aesthetically pleasing bar plots to show slot usage.

3.Occupancy Calculation:

* Iterative Assignment: An iterative approach was used to assign parking slots to vehicles based on their entry and exit times, ensuring that no slot is double-booked.

4.Data Export:

* Excel Output: Exporting processed data and summary statistics to Excel files using Pandas for further analysis or reporting.

5.Summary Metrics:

* Peak Times Calculation: Identifying peak occupancy times and the times with the most available slots.
* These methods and tools help in efficiently analyzing and visualizing the parking data to extract meaningful insights and trends.

## RESULTS AND DISCUSSION :

Results Summary

1.Slot Usage Over Time:A histogram was plotted showing the total occupied time (in hours) for each parking slot.The x-axis represents the slot numbers, and the y-axis shows the total occupied time in hours.This gives a clear view of which slots were used the most.

2.Parking Lot Occupancy Over Time:A line plot was created showing the occupancy over time, with separate lines for the number of occupied slots and available slots.The plot also includes a dashed line representing the total number of slots.This plot helps in understanding the fluctuation in the parking lot occupancy throughout the day.

3.Excel Files:Two Excel files were saved:

Occupancy\_data.xlsx contains the occupancy data for each time slot.

Slot\_usage\_summary.xlsx contains a summary of the slot usage frequency.

4.Peak Occupancy:The peak occupancy (time when the parking lot was most full) and the time when most slots were empty were calculated and printed.The maximum number of available slots at peak occupancy was also calculated.

Interpretation

1.Slot Usage:Slots with higher occupied times are in more demand. This can be due to their location or convenience.Identifying the most used slots can help in optimizing the parking layout or pricing strategy.

2.Occupancy Trends:The occupancy over time plot shows the busiest and least busy times of the day.This information can be used for better resource allocation, such as staffing or maintenance schedules.

3.Peak Times:Knowing the exact times of peak occupancy and when most slots are empty helps in planning and managing the parking lot more efficiently.For instance, promotions or discounts can be offered during off-peak hours to increase usage.

4.Availability:Tracking the availability of slots helps in understanding if the current capacity is sufficient or if there’s a need for expansion.

## CONCLUSION :

Summary of Findings and Future Work:

Findings:

1.Slot Usage Patterns:Identified which parking slots are most and least frequently used.Highlighted times of day with high and low demand.

2.Occupancy Trends:Visualized fluctuations in parking lot occupancy throughout the day.Noted peak times of high occupancy and periods with more available slots.

Future Work:

1.Optimization Strategies:Implement dynamic pricing based on demand to maximize revenue.Adjust staffing levels during peak times to improve customer service.

2.Enhanced Data Collection:Integrate sensors for real-time occupancy monitoring.Expand dataset to include seasonal and day-of-week variations for more comprehensive analysis.

3.Customer Experience Improvements: Develop a mobile app for real-time parking availability updates.Introduce pre-booking options to manage peak demand periods effectively.

CHAPTER 2

## PROGRAM CODE :

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

# Load the data from the Excel file

File\_path = ‘/content/vehicle details.xlsx’

Df = pd.read\_excel(file\_path)

# Processing and Sorting of data

Df[‘Entry Time’] = pd.to\_datetime(df[‘Entry Time’], format=’%H:%M:%S’)

Df[‘Exit Time’] = pd.to\_datetime(df[‘Exit Time’], format=’%H:%M:%S’)

Df.sort\_values(by=’Entry Time’, inplace=True)

# Total number of parking slots available

Total\_slots = 25

# Creating time index and data frame to store occupancy data

Time\_index = pd.date\_range(start=df[‘Entry Time’].min(), end=df[‘Exit Time’].max(), freq=’30T’)

Occupancy\_data = pd.DataFrame(0, index=time\_index, columns=range(1, total\_slots + 1))

# Calculating occupancy and assigning slots

Slot\_assignments = {}

For \_, row in df.iterrows():

Entry\_time = row[‘Entry Time’]

Exit\_time = row[‘Exit Time’]

For slot in range(1, total\_slots + 1):

If slot not in slot\_assignments:

Slot\_assignments[slot] = []

Is\_free = all(not (entry <= entry\_time <= exit or entry <= exit\_time <= exit) for entry, exit in slot\_assignments[slot])

If is\_free:

Slot\_assignments[slot].append((entry\_time, exit\_time))

Occupancy\_data.loc[entry\_time:exit\_time, slot] += 1

Break

# Creating summary DataFrame for slot usage frequency

Slot\_usage\_summary = occupancy\_data.sum(axis=0).reset\_index()

Slot\_usage\_summary.columns = [‘Slot Number’, ‘Total Occupied Time (minutes)’]

Slot\_usage\_summary[‘Total Occupied Time (hours)’] = slot\_usage\_summary[‘Total Occupied Time (minutes)’] / 60

# Plotting the histogram for slot usage

Plt.figure(figsize=(12, 6))

Sns.barplot(x=’Slot Number’, y=’Total Occupied Time (hours)’, data=slot\_usage\_summary, color=’blue’)

Plt.title(‘Slot Usage Over Time’)

Plt.xlabel(‘Slot Number’)

Plt.ylabel(‘Total Occupied Time (hours)’)

Plt.grid(True)

Plt.tight\_layout()

Plt.show()

# Calculating and plotting the occupancy data

Occupancy = occupancy\_data.sum(axis=1)

Available\_slots = total\_slots – occupancy

Plt.figure(figsize=(12, 6))

Plt.plot(occupancy.index.strftime(‘%H:%M’), occupancy, label=’Occupancy’, color=’blue’)

Plt.plot(occupancy.index.strftime(‘%H:%M’), available\_slots, label=’Available Slots’, color=’green’)

Plt.axhline(total\_slots, color=’red’, linestyle=’—‘, label=’Total Slots’)

Plt.fill\_between(occupancy.index.strftime(‘%H:%M’), 0, occupancy, color=’blue’, alpha=0.3)

Plt.title(‘Parking Lot Occupancy Over Time’)

Plt.xlabel(‘Time’)

Plt.ylabel(‘Number of Slots’)

Plt.legend()

Plt.grid(True)

Plt.tight\_layout()

Plt.show()

# Saving the occupancy and slot usage summary data to Excel files

Occupancy\_data.to\_excel(‘occupancy\_data.xlsx’, index=True)

Slot\_usage\_summary.to\_excel(‘slot\_usage\_summary.xlsx’, index=False)

# Printing peak occupancy and maximum available slots

Print(f”Maximum no. Of slots available: {total\_slots}”)

Print(f”Slots Available at Peak occupancy: {available\_slots.min()}”)

# Calculating the total occupancy and total available slot

Total\_occupancy = occupancy\_data.sum(axis=1)

Total\_available\_slots = total\_slots – occupancy\_data.sum(axis=1)

# Finding the time with maximum occupancy and minimum occupancy

Peak\_occupancy\_time = total\_occupancy.idxmax()

Peak\_empty\_slots\_time = total\_available\_slots.idxmax()

Print(“Time at which most slots are filled:”, peak\_occupancy\_time)

Print(“Time at which most slots are empty:”, peak\_empty\_slots\_time)

## USE OF LIBRARIES AND TOOLS :

1.Pandas:Loading and processing the Excel data.Sorting the data by entry time.Creating and managing the occupancy data DataFrame.Summarizing slot usage frequency.

2.Matplotlib:Plotting the slot usage histogram.

Plotting the parking lot occupancy over time.Customizing plots with titles, labels, legends, and grids.

3.Seaborn:Enhancing the bar plot for slot usage with a modern and aesthetically pleasing design.

4.Excel:Storing processed data and results in Excel files for further analysis and sharing.

These libraries and tools facilitate data manipulation, visualization, and storage, making it easier to derive insights and present results clearly.

## IMPLEMENTATION OF ALGORITHMS:

1.Data Sorting:

* Entries are sorted by ‘Entry Time’ to facilitate chronological processing.

2.Slot Assignment Algorithm:

* Iterates over each vehicle’s entry and exit times.
* Checks each slot to see if it is free by ensuring no overlap with existing bookings.
* Assigns the vehicle to the first available slot and marks the occupancy in a DataFrame.

3.Occupancy Calculation:

* Creates a time-indexed DataFrame to track the number of occupied slots at each time interval.
* Updates the occupancy count for each time slot a vehicle is parked.

4.Summary Statistics:

* Calculates the total occupied time for each slot.
* Identifies peak occupancy times and times with the most available slots.

These algorithms efficiently allocate parking slots, track occupancy over time, and summarize usage, providing clear and actionable insights from the data.

# CHAPTER 3

## SOLUTION FEATURE:

## DATA PREPROCESSING :

Data preprocessing refers to the preparation of raw data into a clean and organized format suitable for analysis. Here’s a concise overview of the process:

1.Data Cleaning: Handle missing values, correct errors, and remove duplicates to ensure data integrity.

2.Data Transformation: Normalize or scale data to a common range, encode categorical variables into numerical representations, and perform feature engineering if needed.

3.Data Integration: Combine multiple datasets if necessary, ensuring consistency and compatibility between different sources.

4.Data Reduction: Reduce dimensionality through techniques like PCA (Principal Component Analysis) or feature selection to focus on relevant data.

5.Formatting: Standardize data formats and structures to facilitate analysis and modeling processes.

Effective data preprocessing is crucial for improving the quality and reliability of analytical results, enabling more accurate insights and decisions from the data.

## VEHICLE MOVEMENT ANALYSIS :

* Data Collection: Gather data on vehicle movements, which may include entry and exit times, locations, speeds, and types of vehicles.
* Data Preprocessing: Clean and preprocess the data to ensure accuracy and consistency. This often includes converting timestamps to datetime format, sorting data chronologically, and handling missing or erroneous entries.
* Visualization: Use plots and graphs to visualize vehicle movements over time. Examples include time series plots, heatmaps showing vehicle densities, or flow diagrams illustrating traffic patterns.
* Pattern Analysis: Identify recurring patterns in vehicle movements, such as peak times of activity, frequent routes taken, or areas of congestion.
* Performance Metrics: Calculate key metrics like average occupancy rates, turnover rates, or peak occupancy times to assess efficiency and utilization of parking lots or road networks.
* Predictive Modeling: Employ predictive models to forecast future vehicle movements based on historical data. This can aid in capacity planning, traffic management, or resource allocation.
* Insights and Recommendations: Draw insights from the analysis to make informed decisions. Recommendations may include optimizing traffic flow, improving parking management strategies, or enhancing infrastructure based on observed patterns.

Overall, vehicle movement analysis helps stakeholders—whether in urban planning, transportation management, or commercial operations—better understand and optimize the utilization of vehicle-related resources.

## PARKING OCCUPANCY MONITORING :

Parking occupancy monitoring involves tracking and analyzing the utilization of parking spaces in real-time or over a period. Here’s a concise overview of the process:

1.Data Collection: Gather data on vehicle entries and exits, typically through sensors, cameras, or manual logging systems.

2.Data Processing: Clean and process the collected data to convert timestamps, identify each vehicle’s duration of stay, and determine which parking slots are occupied at any given time.

3.Visualization: Visualize parking occupancy data using graphs, heatmaps, or occupancy charts to show the utilization of parking spaces over time.

4.Analysis: Analyze patterns in parking occupancy to understand peak usage times, average occupancy rates, and trends in parking behavior.

5.Real-time Monitoring: Implement systems for real-time monitoring of parking availability, providing up-to-date information to users or management systems.

6.Optimization: Use insights from occupancy data to optimize parking management strategies, such as adjusting pricing based on demand, improving signage for better navigation, or expanding parking capacity where needed.

7.Reporting and Decision Making: Generate reports and use data-driven insights to make informed decisions about parking policies, infrastructure investments, and operational improvements.

Parking occupancy monitoring helps cities, businesses, and institutions efficiently manage parking resources, reduce congestion, enhance user experience, and optimize revenue generation.

## VEHICLE MATCHING :

Vehicle matching refers to the process of identifying and linking records of the same vehicle across different datasets or instances. Here’s a concise overview:

1.Data Integration: Gather datasets containing vehicle information, such as registration numbers, VINs (Vehicle Identification Numbers), or other unique identifiers.

2.Matching Criteria: Define criteria for matching vehicles, which may include exact matches on identifiers or fuzzy matching based on similarities in attributes.

3.Matching Algorithms: Implement algorithms like string matching, clustering, or machine learning models to identify and link records that likely represent the same vehicle.

4.Data Deduplication: Remove duplicate or redundant vehicle records to ensure data accuracy and integrity.

5.Application: Use matched vehicle data for various applications, such as traffic analysis, insurance claims processing, vehicle history reports, or personalized customer services.

Vehicle matching is crucial for maintaining accurate and consolidated records across systems, enabling efficient data analysis and decision-making processes.

## INSIGHT GENERATION :

Insight generation involves extracting meaningful and actionable information from data. Here’s a concise overview of the process:

1.Data Collection: Gather relevant data from various sources, ensuring it’s clean, structured, and ready for analysis.

2.Data Exploration: Explore the data to understand its characteristics, trends, and relationships. Techniques include visualization, summary statistics, and exploratory data analysis (EDA).

3.Analysis Techniques: Apply statistical methods, machine learning algorithms, or data mining techniques to uncover patterns, correlations, and anomalies within the data.

4.Insight Identification: Identify key findings and insights that provide valuable understanding or solve specific problems. These insights should be relevant, reliable, and actionable.

5.Visualization and Communication: Present insights visually through charts, graphs, or dashboards to facilitate understanding and decision-making among stakeholders.

6.Application: Use generated insights to drive strategic decisions, optimize processes, improve performance, or innovate in business operations.

Insight generation is fundamental for leveraging data-driven approaches to gain competitive advantages, improve efficiency, and enhance decision-making across various domains.

# CHAPTER 4

## TEAM CONTRIBUTION:

Contribution by each member:

Member 1-Vinay M

Data handling and processing :Some of the Vehicle plates images are collected from Google and some are taken in college campus and converted to excel file along with entry time and exit time.Then converted entry time and exit time columns into data time and created a time index and initial occupancy dataframe.Python code is build for the above data.This code differentiates authorised and unauthorised vehicles.

Member 2-Tejashree K

Model development and implementation :Developed further code for calculating occupancy and assigning slots. Summary DataFrame for slot usage frequency is created and histogram for slot usage is plotted.The above data is saved to excel files and time with maximum occupancy and minimum occupancy is estimated.

Member 3-Shravani S

Evaluation of results and report writing : Evaluated and analysed the results generated by the model.Efficiency of output is interpreted and report with detailed information and with proper conclusion of result is prepared.