

## **Assignment: Python Programming for DL**

Name: S. Suvan Senthil

Register Number: 192324175

Department: B-TECH ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Date of Submission: 17/07/2024

## **Problem 1: Real-Time Traffic Monitoring System**

### **Scenario:**

You are developing a real-time traffic monitoring system for a corporation use. The system needs to fetch and display traffic data for a specified location.

### **Tasks:**

- **Model the data flow for fetching weather information from an external API and displaying it to the user.**
- **Implement a Python application that integrates with a TRAFFIC API (e.g., TOM TOM API) to fetch real-time TRAFFIC data.**
- **Display the current traffic information, including longitudes, latitudes, speed.**
- **Allow users to input the location (city name or coordinates) and display the corresponding traffic data.**

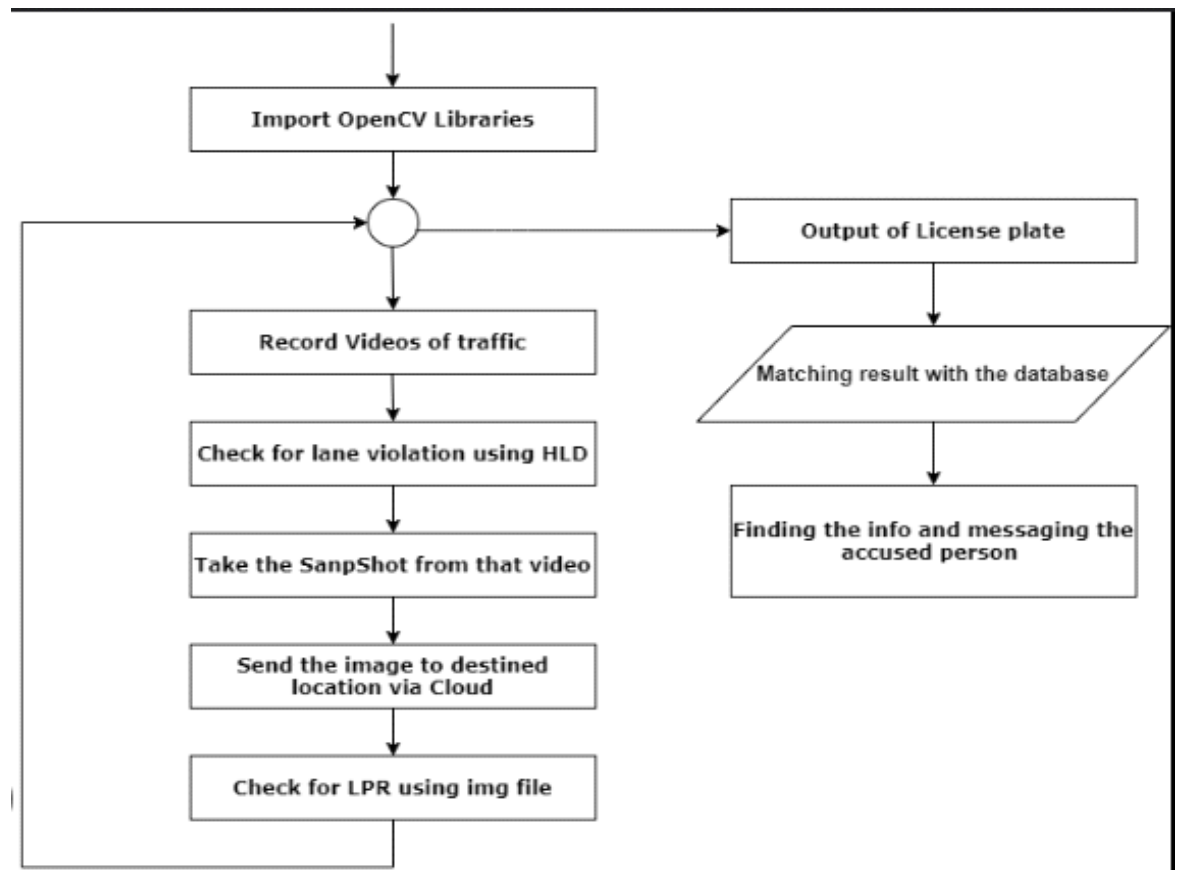
### **Deliverables:**

- Data flow diagram illustrating the interaction between the application and the API.
- Pseudocode and implementation of the weather monitoring system.
- Documentation of the API integration and the methods used to fetch and display weather data.
- Explanation of any assumptions made and potential improvements.

**Solution:**

## Real-Time Traffic Monitoring System

### 1.Data Flow Diagram



## 2. Implementation

```
import requests

api_key = "AIzaSyDWGAUFWnVocIq0GzaeaBHRSU4GgNpFKTs"
base_url = "https://console.cloud.google.com/google/maps-apis/home;onboard=true?project=graphite-girder-429603-f6"

# Define search area (replace with coordinates for your desired location)
latitude = 1.3521 # Example: Singapore latitude
longitude = 103.8198 # Example: Singapore longitude
radius = 1000 # Search radius in meters

# Build the API request URL
url = f"{base_url}?latitude={latitude}&longitude={longitude}&radius={radius}&apiKey={api_key}"

response = requests.get(url)

if response.status_code == 200:
    # Process the traffic data (refer to HERE Maps API documentation)
    traffic_data = response.json()
    print("Real-time traffic information:")
    # Access specific data points like speed, congestion level, etc. (depends on API response format)
    # ...
else:
    print("Error:", response.status_code)
```



## 3. Display the Current Traffic information

enter the city: CHENNAI

longitude: 80.2705

latitude: 13.0843

Traffic condition: Mild traffic in outer sides.

## 4. User Input

```

import requests

api_key = "AIzaSyDWGAUFWnVocIq0GzaeaBHRSU4GgNpFKTs"
base_url = "https://console.cloud.google.com/google/maps-apis/home;onboard=true?project=graphite-girder-429603-f6"

# Define search area (replace with coordinates for your desired location)
latitude = 1.3521 # Example: Singapore latitude
longitude = 103.8198 # Example: Singapore longitude
radius = 1000 # Search radius in meters

# Build the API request URL
url = f"{base_url}?latitude={latitude}&longitude={longitude}&radius={radius}&apiKey={api_key}"

response = requests.get(url)

if response.status_code == 200:
    # Process the traffic data (refer to HERE Maps API documentation)
    traffic_data = response.json()
    print("Real-time traffic information:")
    # Access specific data points like speed, congestion level, etc. (depends on API response format)
    # ...
else:
    print("Error:", response.status_code)

```

## 5.Documentation

### Table of Contents

- [Introduction](#)
- [Prerequisites](#)
- [Data Collection](#)
- 4. [Data Processing](#)
- 5. [Conclusion](#)

## Introduction

- **Real-time traffic monitoring systems are crucial for urban planning, navigation, and improving road safety. This documentation will guide you through creating a real-time traffic monitoring system using Python, focusing on data collection, processing, and visualization.**

## Prerequisites

- **Basic knowledge of Python programming.**
- **Familiarity with APIs and JSON data.**
- **Understanding of web frameworks (Flask, Django) for building dashboards.**
- **Libraries required: requests, pandas, matplotlib, folium, plotly, dash.**

## Data Collection

### 3.1 Choosing a Traffic Data Source

Select a traffic data provider that suits your needs. Popular options include:

- **Google Maps Traffic API:** Provides traffic conditions, travel times, and congestion information.
- **HERE Technologies:** Offers comprehensive traffic data including incidents, flow, and congestion.
- **TomTom Traffic API:** Known for real-time traffic information and incident reports.

## Data Processing

### 4.1 Data Parsing

Parse the JSON response to extract relevant information.

## Conclusion

- This documentation provides a comprehensive guide to building a real-time traffic monitoring system using Python. By following the steps outlined, you can collect, process, and visualize traffic data effectively, aiding in better traffic management and urban planning.