ARTIFICIAL INTELLIGENCE

TRAFFIC LIGHT CONTROL SYSTEM

PROJECT REPPORT

BACHELORS IN TECHNOLOGY

COMPUTER SCIENCE ENGINEERING(AI)

Submitted By:

Ayush Goyal

CSEAI

202401100300083

TABLE OF CONTENT

INTODUCTION

Background

Traffic congestion is a growing issue in urban cities, causing delays, increased fuel consumption, and pollution. Traditional traffic light systems operate on fixed cycles and fail to adapt to real-time conditions, leading to inefficient traffic management.

Objective

This report presents an **AI-driven Traffic Light Control System** that dynamically adjusts signal timings based on traffic density. By integrating real-time data collection, the system ensures smooth traffic flow and minimizes delays.

Scope

- **Automated Traffic Data Collection**
- **Real-time Signal Adjustment**
- **AI-based Optimization Algorithm**
- **Simulation of Traffic Light Execution**

METHODOLOGY

Traffic Data Collection

The system gathers real-time vehicle count data for each lane (North, South, East, and West). In this simulation, traffic is randomly generated, but in real-world scenarios, sensors or cameras would be used.

Signal Timing Adjustment

The system identifies the busiest lane and adjusts the green light duration using the
formula: Green Time=min(60,max(15,Vehicle Count×1.5))Green Time=min(60, max(15,Vehicle Count×1.5)) This ensures that no lane waits too long while optimizing traffic flow.

Traffic Signal Execution

- The green light is allocated to the busiest lane for the calculated duration.
- A **red signal transition** follows before switching to the next cycle.
- The process repeats continuously to adapt to traffic changes.

CODE

```
import time
import random

def get_traffic_data():
    """
    Simulates real-time vehicle count per lane.
    Returns a dictionary with lane-wise vehicle counts.
    """
    return {
        'North': random.randint(5, 50),
        'South': random.randint(5, 50),
        'East': random.randint(5, 50),
        'West': random.randint(5, 50)
    }

def adjust_signal_timing(traffic_data):
    """
    Adjusts green light duration based on vehicle density.
    """
    max_traffic = max(traffic_data, key=traffic_data.get)
    green_time = min(60, max(15, traffic_data[max_traffic] * 1.5))
    return max_traffic, green_time

def traffic_light_control():
```

```
    """
    Runs the AI-based traffic light control system.
    """
    while True:
        traffic_data = get_traffic_data()
        max_lane, green_time = adjust_signal_timing(traffic_data)

        print("\nTraffic Data:", traffic_data)
        print(f"Green Light for {max_lane} lane for {green_time} seconds")
        time.sleep(green_time)  # Simulating the green signal duration
        print("Switching lights...")
        time.sleep(5)  # Red signal duration before switching

# Run the Traffic Light Control System
if __name__ == "__main__":
    traffic_light_control()
```

OUTPUT SCREENSHOT

```
Traffic Data: {'North': 48, 'South': 10, 'East': 47, 'West': 8}
Green Light for North lane for 60 seconds
Switching lights...

Traffic Data: {'North': 40, 'South': 29, 'East': 24, 'West': 24}
Green Light for North lane for 60 seconds
Switching lights...

Traffic Data: {'North': 38, 'South': 40, 'East': 35, 'West': 21}
Green Light for South lane for 60 seconds
Switching lights...

Traffic Data: {'North': 5, 'South': 29, 'East': 25, 'West': 41}
Green Light for West lane for 60 seconds
Switching lights...
```

CONCLUSION

Results and Analysis

The system successfully:

- **Identifies the busiest lane dynamically**.
- **Allocates adaptive green light duration**.
- **Reduces unnecessary wait times**.
- **Optimizes traffic flow efficiency**.

This simulation demonstrates how AI can improve **urban traffic management**, reducing congestion and pollution.

Performance Metrics

- **Response Time:** Real-time adaptation to traffic conditions.
- **Efficiency:** Improved traffic flow compared to fixed signal timers.
- **Scalability:** Can be extended with real-world sensors and cameras.

Future Enhancements

To improve this AI Traffic Light Control System, future developments may include:

- Integration with real-world sensors for accurate traffic data.
- Machine Learning models to predict traffic congestion patterns.
- Adaptive Traffic Prioritization for emergency vehicles.
- Graphical User Interface (GUI) for visualization.

This AI-based traffic light control system efficiently manages traffic flow by dynamically adjusting green light durations based on real-time vehicle counts. By implementing such a system in **smart cities**, authorities can optimize urban mobility, reduce fuel consumption, and lower emissions. Further enhancements using **AI, IoT, and predictive analytics** will make future traffic systems even smarter.