AI POWERED TRAFFIC LIGHT CONTROL SYSTEM

NAME – VINAY KUMAR

BRANCH – CSE AI

SECTION - D

ROLL NO – 2024011003002 79

Report

AI-Powered Traffic Light Control Systems

Introduction

Traffic congestion is one of the most pressing challenges faced by modern cities. Traditional traffic light systems, operating on fixed schedules, fail to adapt to rapidly changing traffic conditions, leading to inefficient traffic flow and increased travel time. AI-powered traffic light control systems are emerging as a transformative technology that leverages artificial intelligence, real-time data, and connected infrastructure to manage urban traffic efficiently. This report delves into the workings, benefits, challenges, and future prospects of AI-based traffic light systems.

Methodology

1. **Data Collection**: Sensors, cameras, and GPS devices gather real-time traffic data like vehicle count, speed, and pedestrian activity.
2. **Data Processing**: Collected data is cleaned, structured, and integrated using IoT for seamless communication across intersections.
3. **AI Model Development**: Machine learning models, including reinforcement learning, are trained on historical and real-time data to optimize traffic flow.
4. **Dynamic Signal Control**: Traffic lights adjust in real-time based on vehicle density, pedestrian needs, and priority for emergency vehicles.
5. **Pilot Testing**: Systems are deployed in controlled environments, evaluated for performance, and refined using feedback.
6. **Optimization and Maintenance**: Regular updates, performance monitoring, and expansion to new intersections ensure scalability.
7. **Stakeholder Collaboration**: Involves city planners, traffic authorities, and public awareness programs for successful implementation.

How AI-Powered Traffic Light Systems Work

1. **Real-Time Data Collection**: Advanced sensors, cameras, and GPS devices capture critical traffic information, such as vehicle count, speed, and pedestrian movement, in real time.
2. **Artificial Intelligence and Machine Learning**:
    - AI algorithms, particularly reinforcement learning, analyze the data to understand traffic patterns and predict congestion points.
    - These models continuously improve over time, becoming more efficient at managing traffic.
3. **Dynamic Signal Optimization**: Traffic signals are adjusted dynamically based on real-time conditions. For example:

- Extending green light durations for high-traffic lanes.
- Quickly switching to red when no vehicles are detected in a particular lane.

4. **Vehicle and Pedestrian Prioritization**:
   - AI systems can prioritize emergency vehicles like ambulances, ensuring quicker passage.
   - Pedestrian signals are optimized for safety and efficiency.
5. **Internet of Things (IoT) Integration**:
   - Connected traffic lights communicate with each other to coordinate and create seamless traffic flow across intersections.
   - Smart vehicles equipped with IoT can interact with the traffic system for further optimization.

Future Prospects

AI-powered traffic light control systems are expected to evolve further with advancements in technology. Future possibilities include:

- **Integration with Autonomous Vehicles**
- **Smart City Ecosystems**
- **Machine Vision and Edge Computing**:

. Conclusion

- AI-based traffic light control systems are a major step ahead in the management of urban traffic. Utilizing artificial intelligence, machine learning, and networked infrastructure, these systems hold enormous promise to improve traffic flow, lower emissions, and increase safety. As cities across the globe increasingly adopt smart technologies, AI-based traffic systems will be instrumental in determining the future of urban mobility.

CODE

```
bcv  import time

import random


# Define traffic light states

RED = "Red"

YELLOW = "Yellow"

GREEN = "Green"


# Timing for each state in seconds

green_time = 30  # Green for 30 seconds

yellow_time = 10  # Yellow for 10 seconds

red_time = 30  # Red for 30 seconds


# Function to simulate traffic light cycle with a fixed total time

def change_traffic_lights(total_time):

    # Initialize the timer for the total cycle time

    timer = 0

    print(f"Total time for this cycle: {total_time} seconds\n")
```

```
    # Start cycling through green, yellow, and red until the total time is reached
    while timer < total_time:
        # Green light phase
        if timer + green_time <= total_time:
            print(f"Green light for {green_time} seconds")
            timer += green_time
            time.sleep(green_time)
        else:
            break  # Exit loop if the green light goes beyond the total time

        # Yellow light phase
        if timer + yellow_time <= total_time:
            print(f"Yellow light for {yellow_time} seconds")
            timer += yellow_time
            time.sleep(yellow_time)
        else:
            break  # Exit loop if the yellow light goes beyond the total time

        # Red light phase
        if timer + red_time <= total_time:
```

```
            print(f"Red light for {red_time} seconds")
            timer += red_time
            time.sleep(red_time)
        else:
            break  # Exit loop if the red light goes beyond the total time

    # Output the final state of the light at the end of the total time
    remaining_time = total_time - timer
    if remaining_time > 0:
        print(f"\nAt the end of {total_time} seconds, the light will still be {RED} for {remaining_time} seconds.")
    else:
        print(f"\nAt the end of {total_time} seconds, the light will be {RED}.")

# Set the total time for the cycle to run (420 seconds)
total_time = random.randint(100, 1000)

# Run the traffic light system for 420 seconds
change_traffic_lights(total_time)
```

OUTPUT/RESULT

```
Total time for this cycle: 313 seconds

Green light for 30 seconds
Yellow light for 10 seconds
Red light for 30 seconds
Green light for 30 seconds
Yellow light for 10 seconds
Red light for 30 seconds
Green light for 30 seconds
Yellow light for 10 seconds
Red light for 30 seconds
Green light for 30 seconds
Yellow light for 10 seconds
Red light for 30 seconds
Green light for 30 seconds

At the end of 313 seconds, the light will still be Red for 3 seconds.
```

CREDIT

Algorithm & Inspiration:

Binary Search Algorithm for efficient guessing

Inspired by AI-driven decision-making techniques

Official Python Website: https://www.python.org/

Tools & Technologies Used:

Programming Language: Python

Development Environment: VS Code / Jupyter Notebook and Microsoft Word and google search engine.

Libraries Used (if any): NumPy, Pandas (if applicable)