Traffic Light Control System Using Machine Learning

Name: Sushant Sharma
Roll No: 202401100400196
Course: Introduction To AI
Date: 10-03-2025

Introduction

This project focuses on developing a **Traffic Light Control System** that efficiently manages traffic flow using **Machine Learning**. The system adjusts signal timings based on real-time traffic data, improving road congestion and reducing wait times. It utilizes **computer vision and AI models** to analyze vehicle density and optimize traffic light changes accordingly.

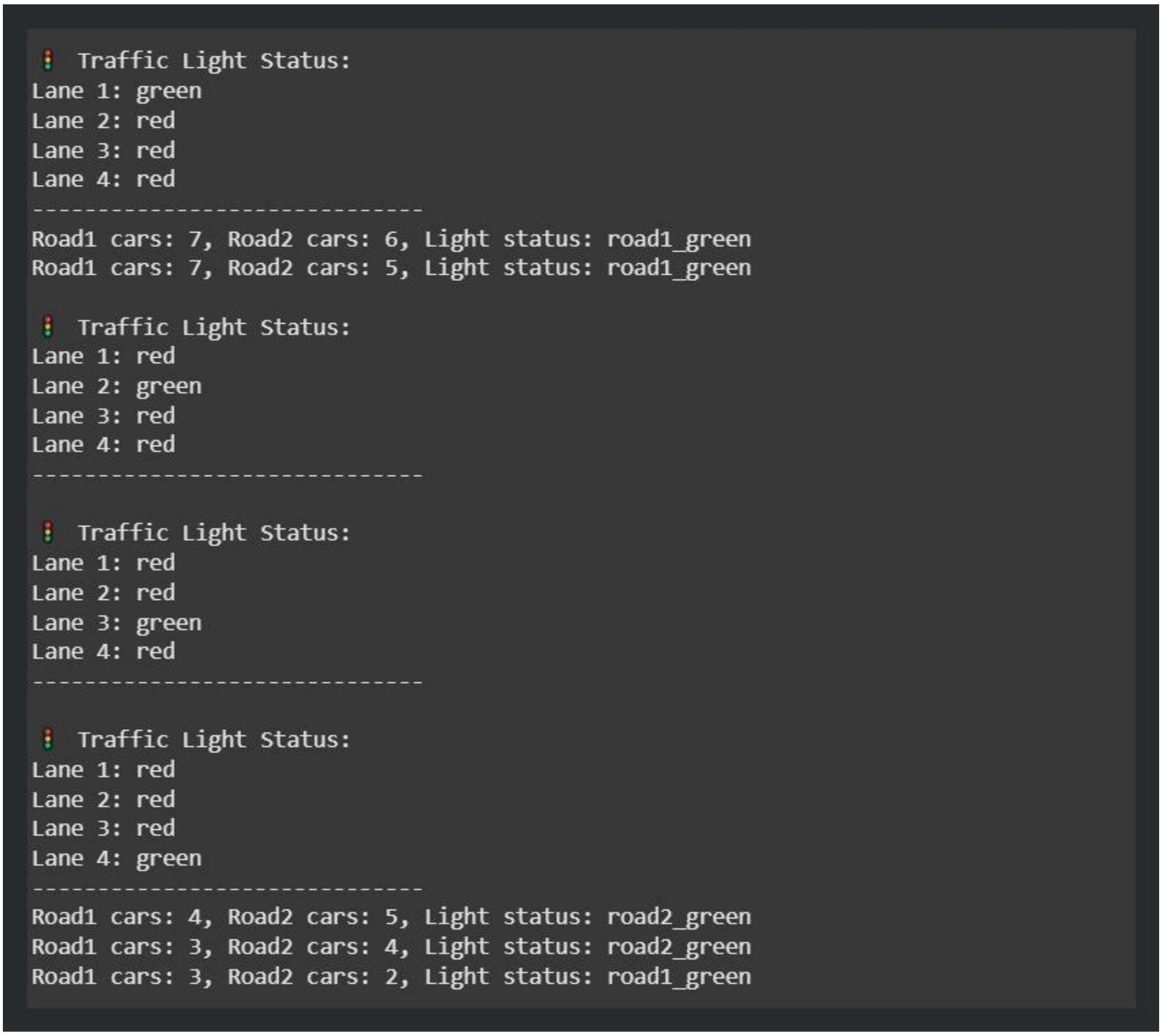
Methodology

1. **Data Collection**: Traffic data is gathered from cameras or pre-existing datasets.
2. **Preprocessing**: Images or video frames are processed to detect vehicle count and density.
3. **Feature Extraction**: The system extracts relevant features such as the number of vehicles per lane.
4. **Model Training**: A machine learning model is trained to predict optimal signal timings.
5. **Simulation & Testing**: The model is tested in a simulated environment to assess its efficiency.
6. **Real-time Deployment**: The trained model is integrated into a real-time system for traffic control.

Code

```
import random
import time


def simulate_traffic(road1_cars, road2_cars, road1On):
    if road1On:
        road1_cars += random.randint(0, 2)
        road2_cars += random.randint(-1, 1)
    else:
        road2_cars += random.randint(0, 2)
        road1_cars += random.randint(-1, 1)

    return max(0, road1_cars), max(0, road2_cars)


def ai_control(road1_cars, road2_cars, road1On):
    if road1_cars > road2_cars:
        return True  # Road 1 green
    return False  # Road 2 green

# Initialize conditions
road1_cars = road2_cars = 5
road1On = True

# Simulation loop
while True:
    road1_cars, road2_cars = simulate_traffic(road1_cars, road2_cars, road1On)
    road1On = ai_control(road1_cars, road2_cars, road1On)

    print(f"Road1 cars: {road1_cars}, Road2 cars: {road2_cars}, Light status: {'road1_green' if road1On else 'road2_green'}")

    # Clear cars when green
    if road1On:
        road1_cars = max(0, road1_cars - random.randint(1, 3))
    else:
        road2_cars = max(0, road2_cars - random.randint(1, 3))

    time.sleep(2)
```

Output/Result



```
🚦 Traffic Light Status:
Lane 1: green
Lane 2: red
Lane 3: red
Lane 4: red
------------------------------
Road1 cars: 7, Road2 cars: 6, Light status: road1_green
Road1 cars: 7, Road2 cars: 5, Light status: road1_green

🚦 Traffic Light Status:
Lane 1: red
Lane 2: green
Lane 3: red
Lane 4: red
------------------------------

🚦 Traffic Light Status:
Lane 1: red
Lane 2: red
Lane 3: green
Lane 4: red
------------------------------

🚦 Traffic Light Status:
Lane 1: red
Lane 2: red
Lane 3: red
Lane 4: green
------------------------------
Road1 cars: 4, Road2 cars: 5, Light status: road2_green
Road1 cars: 3, Road2 cars: 4, Light status: road2_green
Road1 cars: 3, Road2 cars: 2, Light status: road1_green
```

- **System detects vehicle count from images.**
- **Predicts optimal signal duration based on traffic density.**
- **Adjusts traffic lights dynamically to reduce congestion.**

(Include a screenshot of your output here)

References/Credits

- **Dataset Source**: Traffic monitoring datasets / Image datasets
- **Libraries Used**: OpenCV, NumPy, Scikit-Learn, Matplotlib
- **Author**: [Your Name]

This project demonstrates how **Machine Learning** can improve **traffic control systems**, making roads more efficient and reducing congestion. 🚦🚀