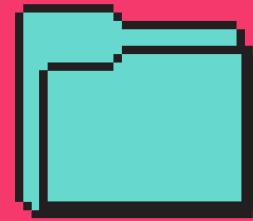


OPTIMIZING TRAFFIC FLOW  
WITH LIVE STREAM FROM TIMES  
SQUARE, NEW YORK



# T5 REAL - TIME TRAFFIC SIGNAL ADJUSTMENT USING YOLO MODEL



**Prepared  
by**

**Name**

Amal

Saad

Nujud

# Table of contents



01 - Introduction



02 - YOLO Model  
Overview



03 - Model Evaluation  
and Results



04 - Deployment



# Introduction

## **Problem Statement:**

**Traffic congestion at intersections is a significant issue, leading to delays.**

**Fixed traffic signal timings do not account for real-time traffic conditions, resulting in inefficient traffic flow.**

## **Solution:**

**We propose a real-time traffic signal adjustment system that uses vehicle detection from live video streams.**

**By analyzing the traffic density in real-time, the system dynamically adjusts the green signal timing based on the number of detected vehicles, improving traffic flow efficiency and reducing congestion.**

**Live Stream Location: Times Square, New York**

**Technology Used: YOLO object detection model for vehicle detection and counting.**



```
def get_stream_url(page_url):

    command = f"streamlink {page_url} best --stream-url"
    result = subprocess.run(command, shell=True, capture_output=True, text=True)

    if result.returncode != 0:
        print(f"Error getting stream URL: {result.stderr}")
        return None
    return result.stdout.strip()

# Function to capture an image from the stream URL
def capture_image(stream_url, output_path):

    cap = cv2.VideoCapture(stream_url)

    if not cap.isOpened():
        print("Error: Could not open video stream.")
        return False

    ret, frame = cap.read()

    if not ret:
        print("Error: Could not read frame from video stream.")
        cap.release()
        return False

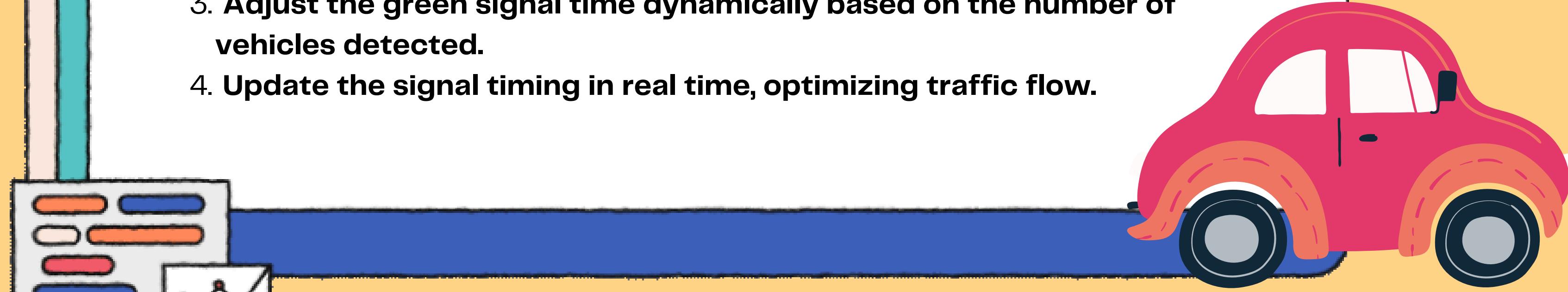
    # Save the captured frame as an image
    cv2.imwrite(output_path, frame)
    cap.release()

    print(f"Image captured and saved as {output_path}")
    return True
```



# YOLO Model Overview

- **Live Video Stream:** Capturing a real-time feed from Times Square, New York, to analyze traffic flow.
- **YOLO Model for Vehicle Detection:** It identifies and classifies vehicles (cars, buses, trucks, and motorcycles.) in each frame of the video stream.
- **Traffic Signal Adjustment Workflow:**
  1. Capture live video frames.
  2. Detect and count vehicles in each frame using the YOLO model.
  3. Adjust the green signal time dynamically based on the number of vehicles detected.
  4. Update the signal timing in real time, optimizing traffic flow.



# YOLO Model Overview

```
def adjust_green_signal_time(vehicle_count):

    base_green_time = 20
    vehicle_multiplier = 2

    green_time = base_green_time + (vehicle_count * vehicle_multiplier)
    return green_time


def process_traffic_stream():
    while capture_video:
        process_frame(frame) # Process frame to detect vehicles
        update_vehicle_count() # Update vehicle count


def adjust_traffic_signal():
    green_time = adjust_green_signal_time(vehicle_count)
    save_green_signal_time(green_time)
```

# YOLO Model Overview

Image captured and saved as captured\_image\_20240905161413.jpg

0: 384x640 3 buss, 8.1ms

Speed: 4.8ms preprocess, 8.1ms inference, 1.5ms postprocess per image at shape (1, 3, 384, 640)

Processed image saved as captured\_image\_20240905161413\_processed.jpg

Vehicle count updated in 'vehicle\_count.txt'.

Total vehicles detected: 3

Adjusted Green Signal Time: 26 seconds

Adjusted green time saved in 'adjusted\_green\_time.txt'.

Image captured and saved as captured\_image\_20240905161420.jpg

0: 384x640 1 bus, 10.6ms

Speed: 2.8ms preprocess, 10.6ms inference, 1.4ms postprocess per image at shape (1, 3, 384, 640)

Processed image saved as captured\_image\_20240905161420\_processed.jpg

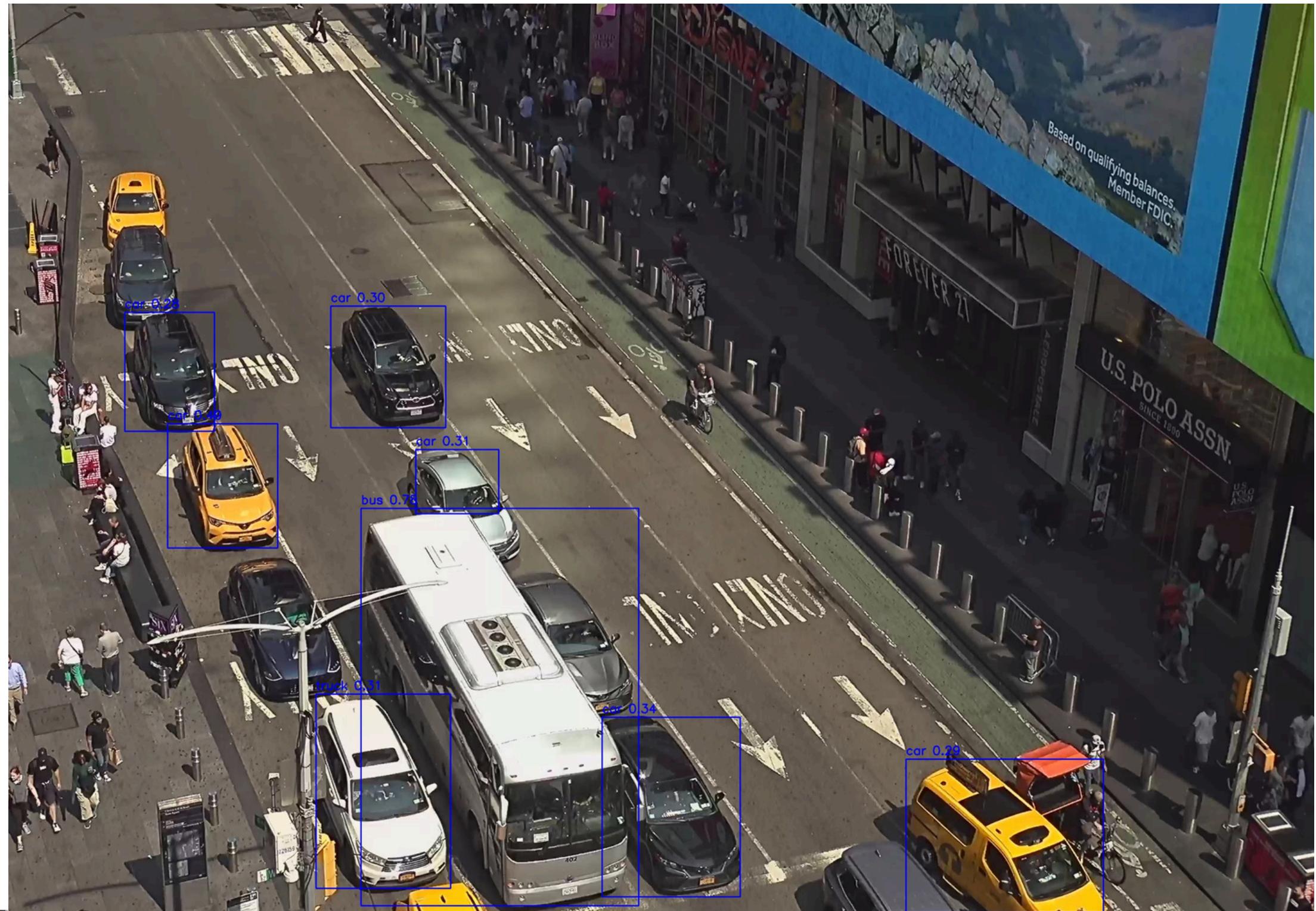
Vehicle count updated in 'vehicle\_count.txt'.

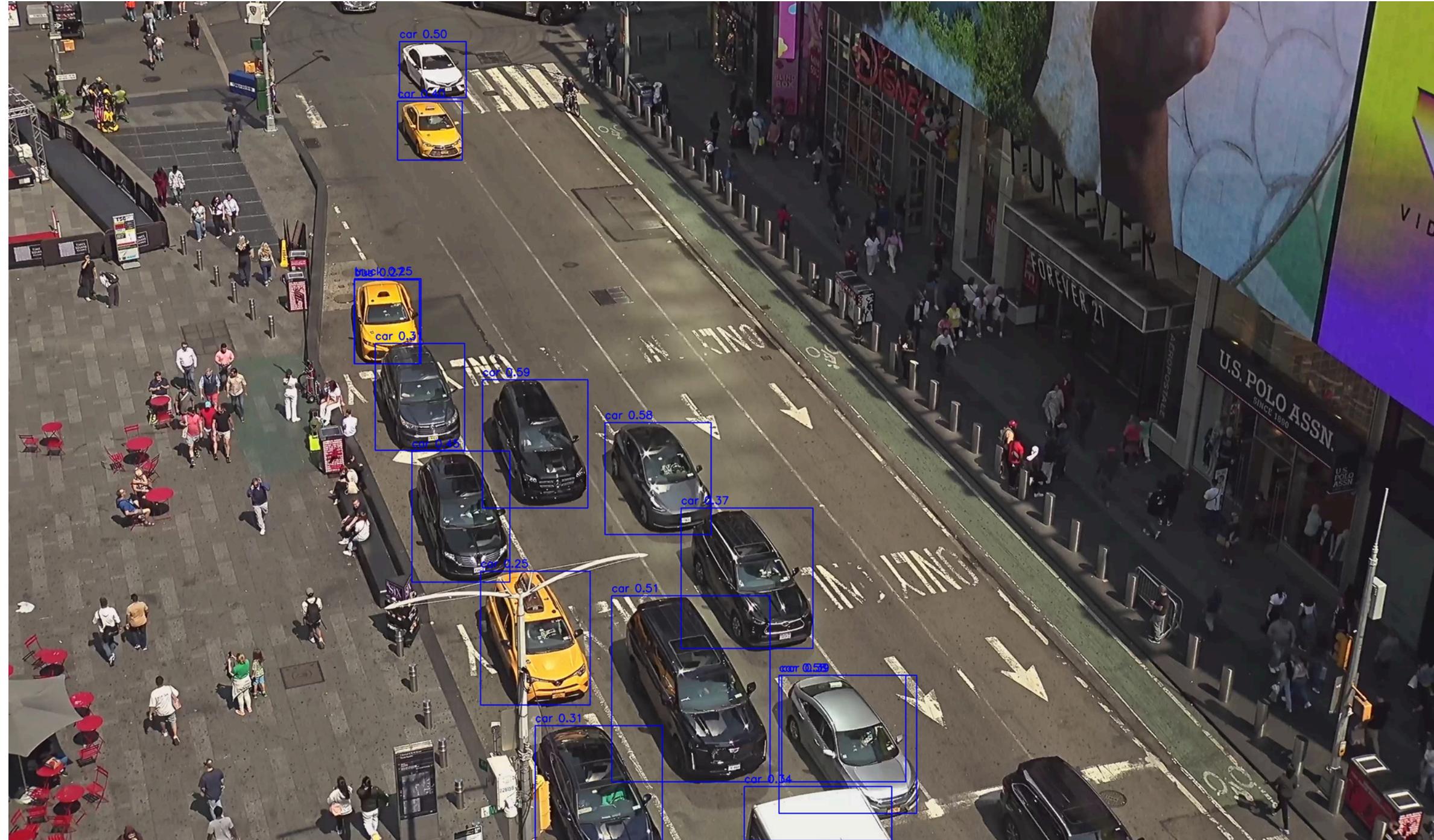
Total vehicles detected: 1

Adjusted Green Signal Time: 22 seconds

Adjusted green time saved in 'adjusted\_green\_time.txt'.

Image captured and saved as captured\_image\_20240905161427.jpg

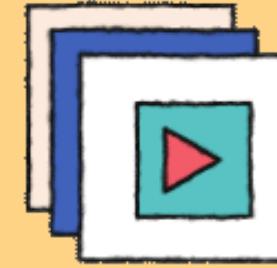






# Model Evaluation and Results

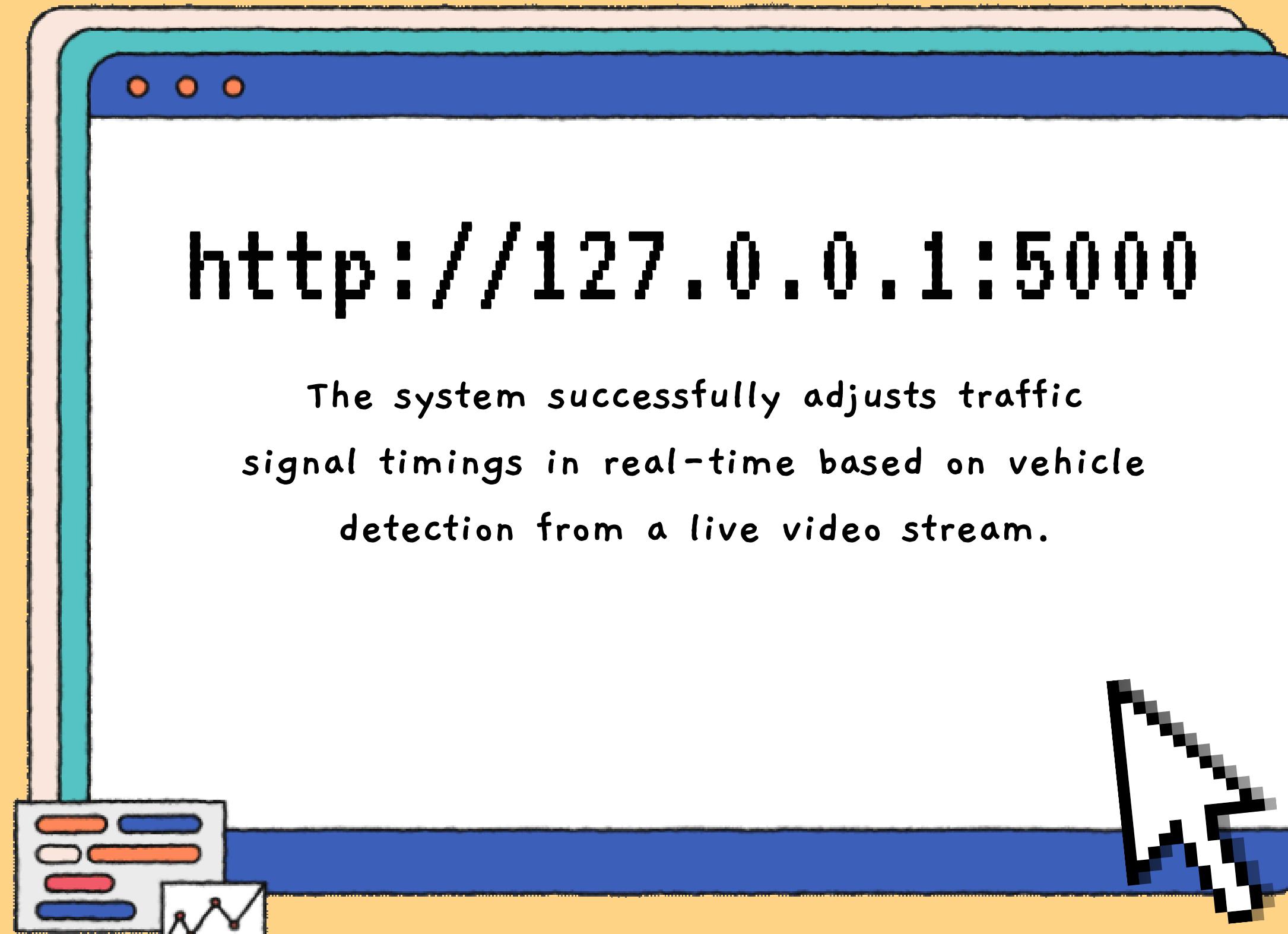
Class	Precision	Recall	mAP50
Overall (All Classes)	81.2	83.7	90.4
Car	81.5	78.5	88.6
Bus	81.0	89.0	92.2

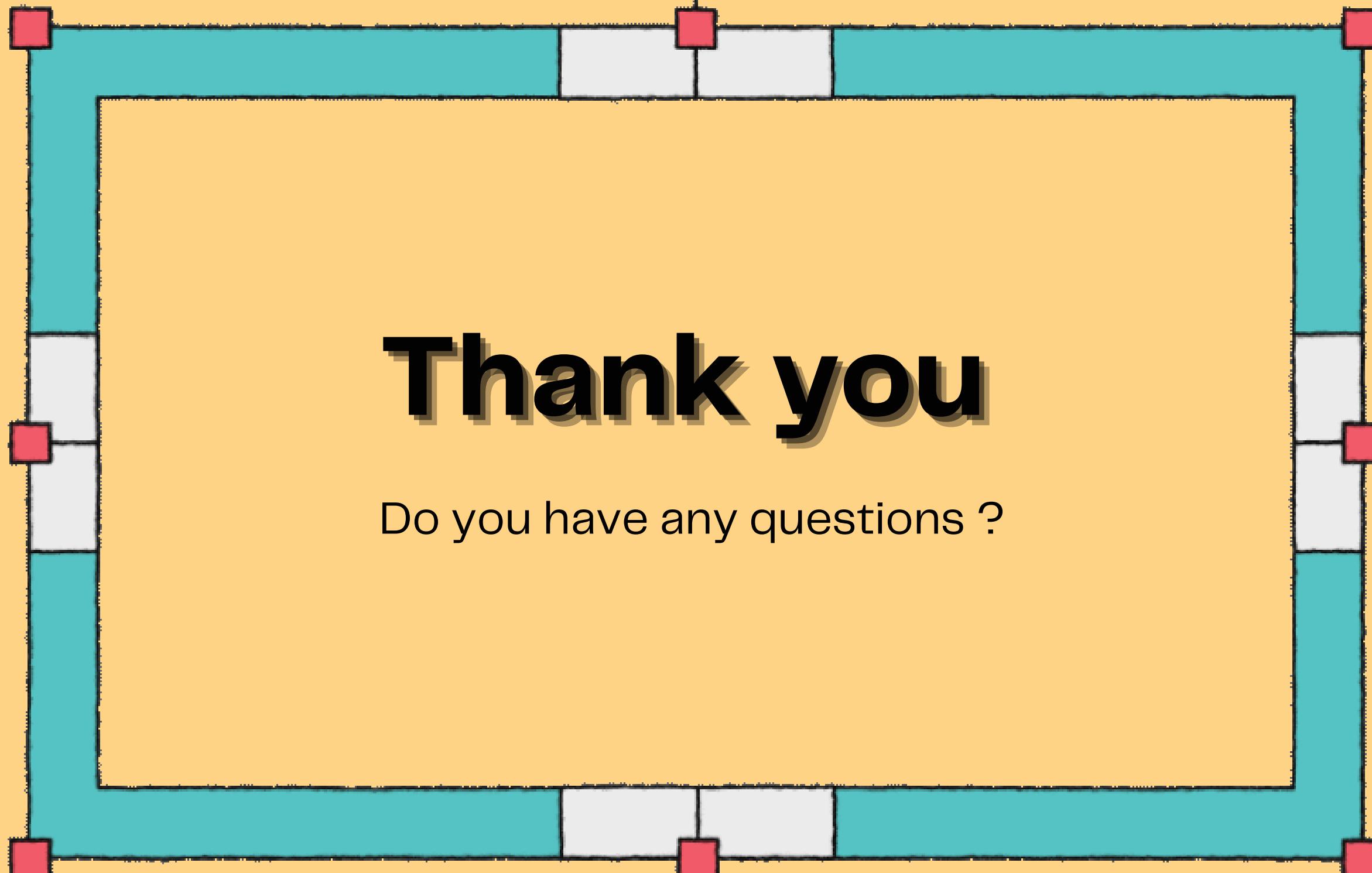


# Deployment

`http://127.0.0.1:5000`

The system successfully adjusts traffic  
signal timings in real-time based on vehicle  
detection from a live video stream.





**Thank you**

Do you have any questions ?