# Case Study Assignment: Smart Traffic Surveillance System

City X has been facing issues with traffic congestion, traffic violations (like running red lights), and inefficient manual surveillance. The municipality plans to deploy an AI-based Smart Traffic Surveillance System using computer vision to:
- Detect vehicles automatically
- Monitor speed
- Identify violations (e.g., wrong turns, red-light jumping)
- Track vehicle movement and congestion levels in real-time

You are part of a data science team tasked with designing and testing a proof-of-concept for this system.

## Objective

Build a mini computer vision system that can:
1. Detect and classify vehicles from traffic video footage or images
2. Track vehicle movement across frames
3. Identify at least one traffic violation (e.g., red light jump)
4. Generate basic statistics like vehicle count, speed estimate, or congestion level

## Dataset Suggestions

Use one of the following:
- AI City Challenge Dataset (Track 1): https://www.aicitychallenge.org/
- Stanford Car Dataset: https://ai.stanford.edu/~jkrause/cars/car_dataset.html
- Open Images Dataset (Vehicle Classes):
https://storage.googleapis.com/openimages/web/index.html
- YouTube dashcam traffic videos or your own recordings

## Background

Discuss recent (2024-25) journal and conference publications on the problem along with the limitations of the models proposed in the publications. State if any of those limitations are addressed in your model.

## Tools & Technologies

Use Python, with libraries such as:
- OpenCV

- YOLOv5 / YOLOv8 for object detection
- DeepSORT or ByteTrack for tracking
- Matplotlib or Seaborn for visualization
Optional: TensorFlow / PyTorch for custom models

## Assignment Tasks

### Part A: Vehicle Detection & Classification

- Use a pre-trained model (e.g., YOLO) to detect vehicles
- Label vehicle types (car, truck, bus, motorcycle)
- Display bounding boxes on sample images or frames

### Part B: Vehicle Tracking

- Implement object tracking across video frames
- Assign consistent IDs to vehicles
- Visualize movement with trajectory lines

### Part C: Violation Detection (Choose One)

- Red Light Violation: Use a static frame with a traffic light zone. If vehicle crosses a virtual line during red signal → flag violation
- Wrong Lane: Flag vehicle entering a restricted lane
- Overspeeding: Use approximate frame difference/time to estimate speed and flag exceeding vehicles

### Part D: Analysis & Reporting

- Count vehicles per minute
- Estimate congestion (e.g., average vehicle density per frame)
- Show sample outputs (images, videos, charts)
- Summarize results in a PDF/Notebook with reflections

## Submission Requirements

1. Jupyter Notebook (well-commented code)
2. 2. Report (PDF/Markdown) with methodology, visuals, and findings (in the form of a conference paper)
3. Demo video

## Evaluation Criteria

| Component | Marks |
| --- | --- |
| Detection Accuracy | 2 |

| | |
|---|---|
| Tracking Consistency | 2 |
| Violation Logic Implementation | 2 |
| Analytical Report Quality | 2 |
| Code Clarity & Reproducibility | 2 |
| Total | 10 |

## Bonus Ideas (Optional)

- Integrate number plate recognition (ANPR)
- Predict traffic flow using historical video patterns
- Use heatmaps to show congestion zones