# Vehicle Attribute Identification and Scene Summary

**Author: Syed Rahmath Ullah Hussaini**

**Guide: Dr. Ahmed Rimaz Faizabadi**

**Institution: Delloyd R&D**

---

**1. Introduction**

In modern traffic monitoring and intelligent transportation systems, vehicle attribute recognition plays a crucial role in ensuring road safety, law enforcement, and traffic flow analysis. This project implements a **computer vision–based system** that detects vehicles in images, identifies their key attributes (type, class, color, and make), and generates a **scene summary** describing the traffic situation.

The solution leverages **YOLOv8 (You Only Look Once, version 8)** for object detection and integrates attribute extraction, license plate recognition, and lane assignment.

---

**2. Problem Statement**

The objective is to design and implement a program that:

1. Detects vehicles and their attributes.

2. Determines whether incoming and/or outgoing traffic is present.

3. Counts the total number of vehicles in an image.

4. Provides detailed per-vehicle information including:

  * Vehicle type/class

  * Color

  * Make / company logo

  * Logo bounding box location

  * Vehicle bounding box location

  * License plate presence and attributes

  * Lane position (left/right)

5. Produces annotated images and structured outputs summarizing the detected scene.

---

**3. Methodology**

### 3.1 Dataset

* Custom dataset prepared with labeled vehicle images containing bounding boxes for vehicles, logos, and license plates.

* Dataset configured in **YOLO format** using a `data.yaml` file.

### 3.2 Model Training

* Base model: **YOLOv8m (medium variant)** from Ultralytics.

* Training configuration:

  * Epochs: 60

  * Batch size: 8

  * Input size: 640 × 640

  * Optimizer: AdamW

  * Learning rate: 0.001

  * Weight decay: 0.0005

  * Augmentations: Mosaic (1.0), Mixup (0.2)

* Hardware: GPU acceleration (CUDA).

The training outputs best weights (`best.pt`) stored inside `runs/detect/…/weights/`.

### 3.3 Inference & Post-processing

During inference, the following steps are performed:

1. **Vehicle Detection** – Detect bounding boxes for vehicles.

2. **Attribute Classification** – For each detected vehicle:

   * **Type/Class**: Car, bus, truck, motorcycle, etc.

   * **Color**: Estimated from pixel color histogram inside bounding box.

   * **Make/Logo**: Identified by detecting manufacturer logos (e.g., Toyota, BMW, Honda).

3. **License Plate Recognition**

   * Check if a license plate is present.

   * Extract bounding box and color.

   * Optionally validate the license number.

4. **Lane Assignment**

   * Divide frame into left and right halves.

  * Classify each vehicle into “Left Lane” or “Right Lane” based on bounding box position.

5. **Traffic Summary**

  * Incoming traffic: Vehicles oriented toward camera.
  * Outgoing traffic: Vehicles oriented away from camera.
  * Vehicle count and per-vehicle details compiled into a JSON-like summary.

### 3.4 Output Generation

* Annotated images: Bounding boxes and labels drawn on frames using OpenCV and PIL.
* Scene summary report generated for each input image.

---

**4. Results**

### 4.1 Annotated Image

* Bounding boxes highlight vehicles, license plates, and logos.
* Labels show type, color, make, and lane.

### 4.2 Scene Summary Example

```json
{
  "IncomingTraffic": true,
"OutgoingTraffic": true,
"TotalVehicles": 3,
"Vehicles": [
  {
    "Type": "Car",
    "Color": "White",
    "Make": "Toyota",
    "LogoBBox": [120, 85, 145, 110],
    "VehicleBBox": [100, 50, 200, 150],
    "LicensePlate": {
      "Present": true,
      "BBox": [130, 120, 180, 140],
      "Color": "Yellow"
    },
    "Lane": "Left"
  },
  {
    "Type": "Truck",
    "Color": "Blue",
    "Make": "Volvo",
    "LogoBBox": [220, 95, 245, 120],
    "VehicleBBox": [200, 70, 320, 180],
    "LicensePlate": {
      "Present": false
    },
    "Lane": "Right"
    }
  ]
}
```

---

## 5. Discussion

* The system successfully integrates **object detection and attribute recognition** into a single pipeline.

* Incoming/outgoing traffic detection depends on orientation of vehicles, which can be improved with additional training data.

* License plate detection and recognition may vary with lighting and resolution.

* Logo detection is challenging when logos are small or occluded.

---

## 6. Conclusion

This project demonstrates a **robust framework** for vehicle attribute identification and scene summarization using YOLOv8. The developed system provides both **visual (annotated images)** and **structured (scene summary JSON)** outputs. Such systems can be extended for real-world applications in:

* Intelligent traffic management.

* Law enforcement and toll systems.

* Automated vehicle surveillance and reporting.

---

## 7. Future Enhancements

* Integrate **OCR** for full license plate recognition.

* Improve **logo classification** using a dedicated CNN.

* Deploy system in **real-time video stream analysis**.

* Use lane detection algorithms instead of simple image division.

---