CAR COLOUR DETECTION
& TRAFFIC ANALYSIS

Internship Report

Submitted by: Tarrush Saxena

Date: December 2025

Project Reference: #Project_Automotive_AI

1. Introduction

This report documents my work on the Car Colour Detection and Traffic Analysis project completed during my internship. The project focuses on building a computer vision system that can detect vehicles in traffic footage and classify them based on their color, specifically identifying blue cars from other vehicles.

The system uses deep learning for object detection combined with traditional computer vision techniques for color classification, creating a practical solution for traffic monitoring and surveillance applications.

2. Background

Traffic management systems have evolved significantly with the integration of AI and computer vision. Modern surveillance requires not just detecting vehicles but also identifying specific attributes like color for tasks such as finding a particular vehicle or analyzing traffic patterns.

The core technologies used in this project include:
- YOLOv8: A state-of-the-art object detection model known for its speed and accuracy
- OpenCV: For image processing and color space transformations
- HSV Color Space: Used for robust color detection under varying lighting conditions
- Python with Tkinter: For building the graphical user interface

3. Learning Objectives

The main learning objectives I set for this project were:

1. Understanding Object Detection: Learn how YOLO architecture works and how to train custom models for vehicle detection.

2. Color Classification: Implement color detection using HSV color space and understand why its better than RGB for this task.

3. Dataset Handling: Work with real-world datasets from Kaggle, including preprocessing and augmentation.

4. GUI Development: Build a functional desktop application using Tkinter that can process video files in real-time.

5. Model Training: Learn the complete ML pipeline from data preparation to training and evaluation.

4. Activities and Tasks

During this project, I completed the following tasks:

Day 1-2: Research and Setup
- Studied YOLOv8 documentation and architecture
- Set up the development environment with required libraries
- Downloaded and explored the Vehicle Detection dataset from Kaggle

Day 3-4: Model Development
- Trained YOLOv8 model on the vehicle dataset for 100 epochs
- Implemented the color classification module using HSV masking
- Created the detection wrapper class for easy integration

Day 5: GUI and Integration
- Built the Tkinter-based GUI with video upload functionality
- Integrated detection and color classification modules
- Added real-time counting display for vehicles and people

Day 6: Testing and Documentation
- Tested the system with various traffic videos
- Created Jupyter notebooks for model evaluation
- Wrote documentation and prepared the final report

5. Skills and Competencies

Through this project, I developed and strengthened the following skills:

Technical Skills:
- Deep Learning: Training and fine-tuning YOLO models
- Computer Vision: Working with OpenCV for image processing
- Python Programming: Writing clean, modular code
- GUI Development: Building desktop apps with Tkinter
- Data Analysis: Using matplotlib for visualization

Soft Skills:
- Problem Solving: Debugging issues with model training and dataset paths
- Time Management: Completing the project within the internship timeline
- Documentation: Writing clear code comments and technical documentation
- Self-learning: Researching solutions independently when stuck

6. Feedback and Evidence

The project was evaluated based on the following metrics:

Model Performance:
- mAP@50: 0.95+ (excellent detection accuracy)
- Precision: High (minimal false positives)
- Recall: High (captures most vehicles in frame)

The training results, confusion matrix, and F1 curves are documented in the notebooks folder. Sample predictions show accurate bounding boxes around vehicles with correct color classification.

Evidence of work:
- Training logs in CarTrafficProject_Train folder
- Evaluation notebooks with visualizations
- Working GUI application (main.py)
- Complete source code with documentation

7. Challenges and Solutions

I encountered several challenges during this project:

Challenge 1: Windows Path Errors
The default dataset directory caused permission errors on Windows. I solved this by configuring a local datasets folder in the project directory.

Challenge 2: False Positives in Color Detection
Initially, the blue detection was catching sky and road colors. I fixed this by focusing on the center region of detected cars and tightening the HSV thresholds to require higher saturation values.

Challenge 3: Model Training Time
Training for 100 epochs took significant time. I used early stopping with patience=20 to prevent overfitting and reduce unnecessary training time.

Challenge 4: GUI Responsiveness
The video processing was freezing the GUI. I solved this by running the detection loop in a separate thread using Pythons threading module.

8. Outcomes and Impact

Project Outcomes:

1. A fully functional car detection and color classification system
2. Trained YOLOv8 model achieving 95%+ mAP on vehicle detection
3. Real-time processing capability through the desktop application
4. Comprehensive documentation and evaluation notebooks

Potential Impact:
- Traffic Monitoring: Can be used to track specific vehicles in surveillance footage
- Parking Systems: Identify and count vehicles by color in parking lots
- Law Enforcement: Aid in finding vehicles matching specific descriptions
- Smart City Applications: Integration with traffic management systems

9. Conclusion

This internship project provided valuable hands-on experience in building a complete computer vision application from scratch. I learned how to work with modern deep learning frameworks, handle real-world datasets, and create user-friendly applications.

The Car Colour Detection system successfully demonstrates the integration of object detection with traditional computer vision techniques. The high accuracy achieved shows that YOLOv8 is an excellent choice for vehicle detection tasks.

Going forward, the system could be enhanced with:
- Support for multiple color detection (not just blue)
- Real-time webcam input
- Integration with database for logging detected vehicles
- Mobile application development

I am grateful for this opportunity to apply theoretical knowledge in a practical project and develop skills that will be valuable in my future career in AI and computer vision.