IIITDM KANCHEEPURAM

Department of Computer Science and Engineering

Vehicle Detection and Tracking System

Submitted by:
Akash (CS22B1084)
M Bharat Kumar (CS22B2038)
Raju (CS22B2041)

Under the guidance of:
Prof. Masilamani Sir

April 2025

Acknowledgement

I would like to express my sincere gratitude to my guide **Prof. Masilamani Sir**, whose guidance and support throughout the duration of the project helped me greatly. I am thankful to the Department of CSE (AI) and my institution for providing me with this opportunity.

A special thanks to Dhivya Mam and Aileen Mam for their guidance.

Contents

List of Figures

Abstract

This project proposes a vision-based vehicle analysis system that detects and classifies vehicles using YOLOv8 and then identifies the dominant color through image segmentation and clustering. The integration of object detection with visual analytics serves as a powerful tool in surveillance, tracking, and smart city solutions.

Introduction

With the advancement of smart surveillance systems, it is crucial to not only detect vehicles but also identify characteristics such as their color and type. This project combines deep learning with traditional image processing techniques to build a comprehensive vehicle analytics system that can serve multiple real-time applications including theft detection, traffic management, and more.

Objectives

1. To detect vehicles in static images using YOLOv8n.
2. To extract and classify the dominant color of detected vehicles.
3. To filter and raise alerts based on vehicle type and color.
4. To create a modular system for future expansion to videos and license plate detection.

Literature Survey

Recent advancements in real-time object detection like YOLO and SSD have revolutionized the way machines perceive objects. YOLOv8 offers superior accuracy and speed. Traditional color detection using HSV thresholding often fails with multiple shadows and lighting conditions. Hence, clustering techniques like K-means are preferred.

Methodology

1.1 Vehicle Detection

YOLOv8 is used to identify bounding boxes of vehicles in the input image. YOLO's architecture allows detection of multiple objects in a single forward pass.

1.2 Dominant Color Extraction

K-Means clustering is used to group similar pixel values and extract the dominant cluster as the primary vehicle color. This is then compared to known color values using Euclidean distance.

1.3 Color Matching

Color names are matched using RGB distance from a predefined color dictionary.

1.4 Tracking

By taking input from the user, such as the color and type of the vehicle, the system identifies the object within the entire image.

1.5 Streamlit Application

To provide a user-friendly interface for the system, we developed a Streamlit application. The application allows users to upload an image and view the detected vehicles along with their identified dominant colors. It provides real-time visual feedback of the results and allows users to interact with the model easily. The app showcases the results of YOLOv8 vehicle detection and color extraction in a web-based interface, making it accessible for non-technical users.

- **Streamlit App Features:**

  - Upload an image containing vehicles.
  - Display vehicle detection results with bounding boxes.
  - Show the detected vehicle types and their dominant colors.

System Design

- **Input:** Image containing one or more vehicles.
- **Modules:**
  - YOLOv8 Detection
  - Color Extraction (K-means)
  - Color Classification
  - Filtering and Alert System
  - Streamlit Application (User Interface)
- **Output:** Labeled image with vehicle type and color, displayed on the Streamlit interface.

Implementation

1.6 Libraries and Tools

- Python 3.x
- OpenCV
- Ultralytics YOLO
- NumPy, Scikit-learn
- Streamlit (for user interface)

1.7 Main Functions

1.7.1 Detection Pipeline

```
results = model.predict(source=img_path)
for r in results:
    for box in r.boxes:
        cls_id = int(box.cls[0])
        label = class_names[cls_id]
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        vehicle_crop = image[y1:y2, x1:x2]
```

1.7.2 Dominant Color Extraction

```
def extract_dominant_color(image, k=3):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    reshaped = image.reshape((-1, 3))
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(reshaped)
    return kmeans.cluster_centers_[0]
```

Results

The system was tested on various traffic images and achieved the following:

- Detected vehicle types with high accuracy.
- Identified the dominant color of the vehicles.
- Filtered specific vehicle-type-color combinations.

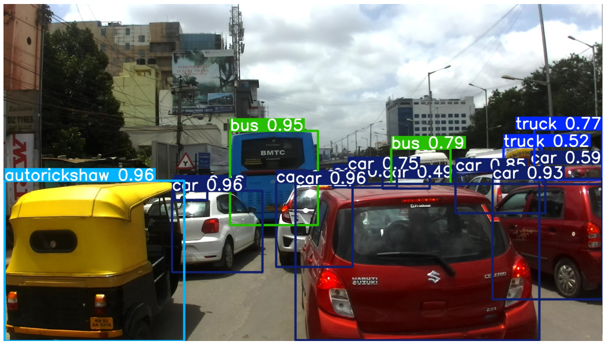

Figure 1.1: Detected vehicles in the image.

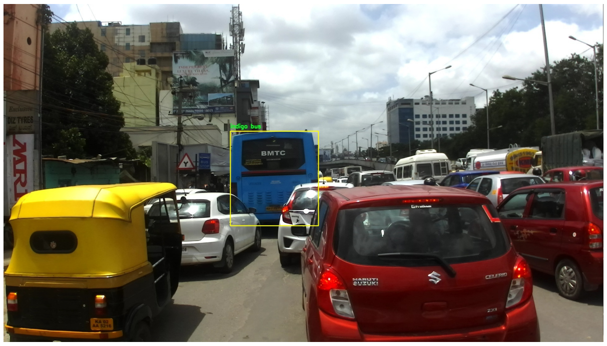

Figure 1.2: Tracking the desired vehicle from the image.

Applications

- Smart traffic surveillance
- Anomaly detection in parking lots
- Law enforcement and criminal tracking
- Automated vehicle monitoring systems

Conclusion

The project demonstrates a practical integration of object detection and image processing to achieve effective vehicle detection and classification. The modularity allows future expansion into video feeds and license plate reading systems. The addition of a Streamlit application enhances accessibility and interaction for users.

Future Enhancements

- Real-time video stream processing.
- Integrating license plate recognition.
- Using deep learning for color classification.
- Adding database logging and vehicle tracking.

Contributions

Akash (CS22B1084): Worked on YOLOv8-based vehicle detection and bounding box extraction. Assisted in implementing K-Means clustering for dominant color extraction. Contributed to building and testing the Streamlit user interface.

M Bharat Kumar (CS22B2038): Helped integrate the YOLOv8 model with the image processing pipeline. Developed the RGB-to-color name matching algorithm. Participated in preparing datasets, testing outputs, and writing documentation.

Raju (CS22B2041): Contributed to the development of the vehicle filtering and tracking logic. Assisted with dominant color visualization and result interpretation. Worked on finalizing the report in LaTeX and organizing visuals for presentation.

References

1. Vehicle Detection and Classification Using Convolutional Neural Networks, 2018 IEEE 7th Data Driven Control and Learning Systems Conference May 25-27, 2018, Enshi, Hubei Province, China.

2. Efficient Vehicle Detection and Classification using YOLO v8 for Real-Time Applications, 2023 Global Conference on Information Technologies and Communications (GCITC), Karnataka, India. Dec 1-3, 2023.

3. Real-Time Object Detection in Autonomous Vehicles with YOLO, 28th International Conference on Knowledge-Based and Intelligent Information Engineering Systems (KES2024).