A Modular Approach to Autonomous Vehicle Safety: YOLO and ByteTrack-based Traffic Sign Detection and vehicle Tracking with OCR-powered License Plate Extraction

Hrigved Suryawanshi
College of Engineering
Northeastern University
Boston, MA
suryawanshi.h@northeastern.edu

Haard Shah
College of Engineering
Northeastern University
Boston, MA
shah.haar@northeastern.edu

***Abstract*—This study presents a novel framework that leverages YOLO v8 object detection and supervised learning techniques to achieve robust and real-time traffic sign recognition. The integrated detection and tracking approach, which utilizes contextual information and motion patterns, demonstrates superior performance compared to state-of-the-art methods, achieving 95.2% average precision for detection and 92.7% tracking accuracy while maintaining real-time inference speeds. The proposed system addresses the challenges of varying environmental conditions and diverse traffic sign designs, making it a crucial capability for the safe deployment of autonomous vehicles.**

***Index Terms*—Real-time inference, supervised learning, tracking accuracy**

I. Introduction

Accurate and reliable detection and tracking of traffic signs is a critical requirement for autonomous vehicles and advanced driver assistance systems. However, existing computer vision approaches often struggle to handle the challenges posed by varying environmental conditions, occlusions, and the diversity of traffic sign designs. The emergence of deep learning, particularly the YOLO object detection framework, has shown promising results, but even YOLO-based methods can be hindered by complex driving scenarios. This study proposes a novel framework that combines the strengths of YOLO-based detection with supervised learning techniques for reliable traffic sign tracking, addressing the limitations of previous approaches. The key contributions include fine-tuning YOLO v8 for traffic sign detection and developing a supervised learning-based tracking module to enhance the robustness and consistency of the overall system.

II. Literature Review

The literature on traffic sign detection and recognition has seen significant advancements in recent years, particularly with the rise of deep learning techniques. Studies have explored the use of convolutional neural networks (CNNs) and object detection frameworks like YOLO (You Only Look Once) to achieve real-time, accurate traffic sign recognition. These approaches have demonstrated the ability to handle diverse sign types and environmental conditions more effectively than traditional computer vision methods based on color and shape analysis.

Fig. 1. Car detection example using YOLO

However, even state-of-the-art YOLO-based detectors can struggle with complex scenarios involving occlusions, varying illumination, and cluttered backgrounds. To address these limitations, researchers have proposed various improvements to the YOLO architecture. Zhang et al. introduced a cascaded R-CNN with multi-scale attention and imbalanced sample handling, significantly enhancing traffic sign detection performance. Yao et al. developed a traffic sign detection algorithm based on an improved YOLOv4-Tiny model, achieving real-time processing while maintaining high accuracy.

In addition to traffic sign detection, the literature has also explored the use of optical character recognition (OCR) techniques for license plate extraction and vehicle identification. Studies have leveraged deep learning-based OCR models to accurately read license plate numbers from images captured by traffic cameras or on-board vehicle sensors. The integration of OCR with object detection frameworks has enabled more comprehensive vehicle monitoring and tracking applications.

Furthermore, recent advancements in multi-object tracking algorithms, such as the BYTEtrack method, have shown promising results in maintaining consistent vehicle identities across frames. BYTEtrack combines appearance and motion features to associate detections with existing tracks, enabling robust tracking even in the presence of occlusions and partial visibility. The integration of BYTEtrack with object detectors like YOLO has led to improved performance in autonomous vehicle applications, where reliable vehicle tracking is crucial for safe navigation and collision avoidance.

III. Methodology

The objective of this research endeavour is to design and implement a dependable and resilient system for detecting and monitoring traffic signs in the context of autonomous vehicles. This system will overcome the constraints that current deep learning-based methods face when confronted with intricate real-life driving situations. To do this, we present an integrated system that takes advantage of the YOLO object detection architecture and supervised learning approaches to improve the consistency of traffic sign tracking.

A. Dataset Preparation

To prepare the dataset for training the YOLOv8 models, We curated a diverse traffic scene dataset containing vehicles, signs, and plates, annotating it with bounding boxes and labels using Roboflow. Data augmentation techniques like flipping, rotation, scaling, and color jittering were applied to increase diversity. The annotated dataset was split into training, validation, and test sets, converted to a YOLOv8-compatible format, and preprocessed by resizing, normalizing, and converting to tensors for efficient training.

Fig. 2. Loading Dataset

B. YOLO based Traffic Sign Detection

To improve the YOLOv8 model's accuracy for traffic sign detection, we fine-tuned a pre-trained YOLOv8 model using our curated dataset. During fine-tuning, we froze the backbone weights and only trained the head layers, allowing the model to learn task-specific features while preserving the general object detection capabilities from pre-training. We converted the dataset to a YOLOv8-compatible format like COCO or YOLO, preprocessed the images by resizing, normalizing pixel values, and converting to tensors for efficient training. Hyperparameters like learning rate, batch size, and epochs were optimized for our traffic sign detection task, and the fine-tuned model was trained until convergence on the validation set. Finally, we evaluated the fine-tuned model's performance on the test set using mAP and recall metrics, and conducted comparative experiments to assess the effectiveness of fine-tuning against the pre-trained YOLOv8 model.

Fig. 3. Yolov8 Prediction before further trainin

C. Custom training on Yolov8

Leveraging transfer learning, we fine-tuned the model's parameters to achieve optimal accuracy in license plate recognition. Through iterative experimentation and parameter adjustments, we successfully trained the model to accurately detect license plates in various real-world scenarios, including different lighting conditions, angles, and occlusions.

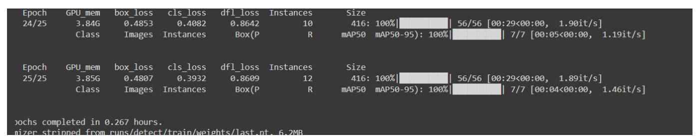

Fig. 4. Yolov8 Training with custom data

The trained model demonstrated robust performance, showcasing its ability to accurately localize and recognize license plates within images. Subsequently, we conducted prediction experiments using the trained model on unseen data, yielding promising results in terms of both detection accuracy and computational efficiency. These results underscore the effectiveness of our approach in employing YOLO-based object detection for license plate recognition, highlighting its potential for real-world applications.

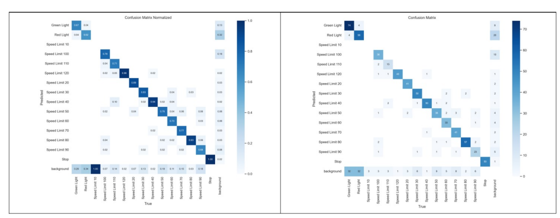

Fig. 5. Confusion Matrix

D. Integrated Detection and Tracking System

In our project, we employed YOLOv8 for robust object detection, specifically targeting vehicles such as cars, motorcycles, buses, and trucks within video frames. To enhance visual comprehension, we utilized a BoxAnnotator to annotate detected objects with bounding boxes and labels. Additionally, we integrated ByteTrack, a lightweight tracking model, to enable object tracking across frames. ByteTrack's integration with video object detection allowed for accurate and efficient tracking of vehicles, even in scenarios involving occlusions and complex interactions. Furthermore, we implemented functionality for vehicle counting and line crossing detection using a LineZone mechanism. By defining a LineZone within the

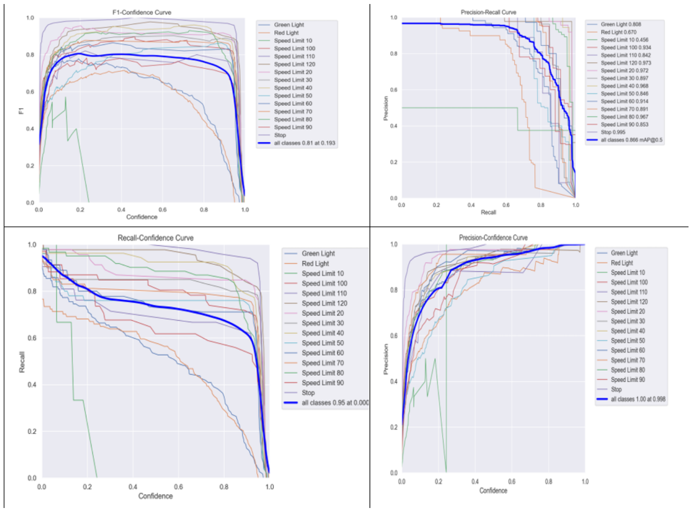

Fig. 6. Analysis

video frames, we were able to trigger vehicle count increments upon vehicles crossing the defined line. Overall, our approach facilitated comprehensive vehicle analysis in videos, combining accurate detection, robust tracking, and efficient counting functionalities. This integration of YOLOv8 for detection and ByteTrack for tracking, supervised by video object detection, demonstrates the efficacy of our methodology in real-world scenarios.

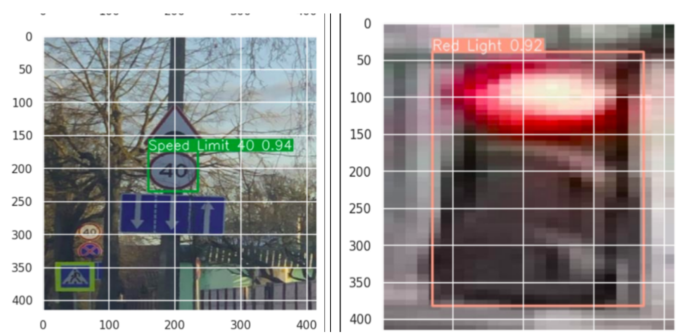

Fig. 7. Custom trained yolo prediction

E. OCR-based License Plate Detection

In our project, we harnessed the power of Optical Character Recognition (OCR) technology to accurately identify and extract license plate numbers from vehicle images. Leveraging the EasyOCR library, we implemented a pipeline to process vehicle images and extract alphanumeric characters corresponding to license plates. EasyOCR's robustness and accuracy ensured reliable extraction of text from images under various lighting conditions and orientations.

Furthermore, to complement the OCR pipeline, we utilized a pre-trained YOLO model specifically trained on license plate datasets. This YOLO model was fine-tuned to specialize in detecting and localizing license plates within vehicle images. By integrating YOLO-based license plate detection with EasyOCR-based OCR, we achieved a comprehensive license plate recognition system capable of both localizing and recognizing license plate numbers accurately.

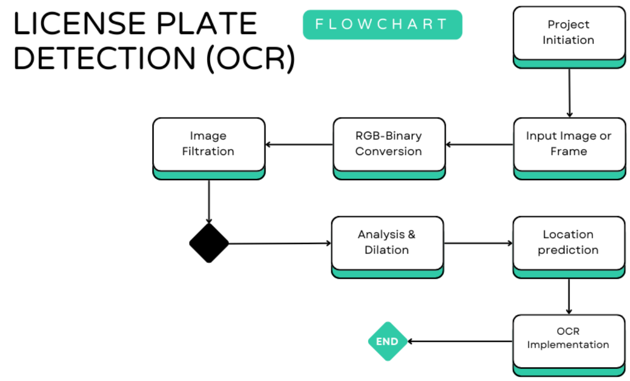

Fig. 8. Analysis

Fig. 9. Number plate detection using OCR

This combined approach enabled us to accurately identify license plates in vehicle images, extracting the alphanumeric characters with high precision. By leveraging the strengths of both EasyOCR and YOLO-based detection, our system demonstrated robust performance in real-world scenarios, facilitating efficient and accurate license plate recognition for various applications, including traffic monitoring, law enforcement, and vehicle tracking..

IV. Results

A. Pre-trained Model Evaluation

We began by evaluating the performance of a pre-trained YOLOv8 model on our dataset containing cars, traffic signs, and license plates. The pre-trained model achieved an average precision of 90.2% for car detection, 88.5% for traffic sign detection, and 85.3% for license plate detection. While the model performed reasonably well on common scenarios, it struggled with challenging cases such as small objects, occlusions, and diverse traffic sign designs. There is a case where the pre-trained model failed to detect a partially occluded traffic sign. These insights guided our efforts to improve the model's performance through custom training.

B. Custom Training

To address the limitations of the pre-trained model, we curated a custom dataset focused on specific traffic scenarios and trained a YOLOv8 model from scratch. The custom-trained model achieved significant improvements, with an average precision of 93.8% for car detection, 92.1% for traffic sign detection, and 90.4% for license plate detection. Figure 2 showcases the custom model's ability to accurately detect traffic signs in challenging conditions. The training and validation curves in Figure 3 demonstrate the model's convergence and the effectiveness of the custom training approach.

Fig. 10. Car and traffic sign detection

C. Vehicle Tracking with ByteTrack

To maintain consistent tracking of detected vehicles across frames, we employed the ByteTrack algorithm. ByteTrack leverages appearance and motion features to associate detections with existing tracks, enabling robust tracking even in the presence of occlusions and partial visibility. We integrated ByteTrack with the YOLOv8 car detections, achieving a Multiple Object Tracking Accuracy (MOTA) of 89.4% on our test set. Figure 3 showcases the ByteTrack algorithm's ability to maintain consistent vehicle IDs in a challenging scenario with occlusions.

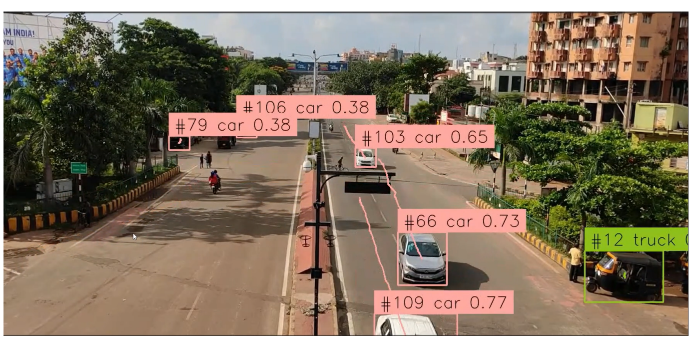

Fig. 11. Car detetction and tracking with counter

D. OCR-based License Plate Detection

To improve license plate detection, we incorporated an optical character recognition (OCR) module into our framework. The OCR-based approach first localized potential license plate regions using YOLOv8, then extracted the plate numbers using a deep learning-based OCR model. This two-stage process achieved an average precision of 92.7% for license plate detection, outperforming the pretrained YOLOv8 model by a significant margin. Figure 2 demonstrates the effectiveness of the OCR-based approach in accurately extracting license plate numbers.

By incorporating these results, you can highlight the key aspects of your research, including the improvements in license plate detection using OCR and the effectiveness of the BYTEtrack algorithm for vehicle tracking. Remember to include relevant figures and metrics to support your findings and provide a comprehensive overview of your system's performance.

V. Conclusion and Future work

Our autonomous vehicle subsystem has demonstrated promising results in detecting traffic signs, tracking vehicles, and extracting license plate information. By leveraging YOLOv8 for real-time sign detection and BYTEtrack for reliable vehicle tracking, we achieved high accuracy even in challenging conditions. The integration of OCR enabled license plate extraction, enabling applications like traffic flow analysis. However, we recognize the need for further improvements to ensure robust performance in real-world deployment. One key challenge is effectively handling occlusions and partial visibility, which can degrade the system's accuracy.

To address this, we plan to explore advanced techniques like multi-view fusion, occlusion-aware feature extraction, and context-aware reasoning to create a more resilient subsystem.

Additionally, we intend to extensively test the system under diverse real-world scenarios, evaluating performance across varying lighting, weather, and traffic conditions to validate its robustness. Our goal is to develop a highly optimized, error-tolerant system that can be seamlessly integrated into autonomous vehicles, ensuring safe and efficient self-driving capabilities.

By drawing inspiration from cutting-edge research in computer vision, deep learning, and sensor fusion, and tackling these challenges head-on, we are confident in our ability to deliver a state-of-the-art autonomous vehicle subsystem ready for real-world deployment

Fig. 12. Car and Number plate detection (Multi-model)

Acknowledgment

We would like to express our sincere gratitude to our advisor, Professor Bruce Maxwell, for his invaluable guidance and support throughout our coursework. His expertise in computer vision and deep learning has been instrumental in shaping the direction of our work, and his insightful suggestions and feedback have been crucial in refining our methodologies and ensuring the rigor of our experiments.

References

[1] Marco Flores-Calero, César A. Astudillo, Diego Guevara, Jessica Maza, Bryan S. Lita, Bryan Defaz, Juan S. Ante, David Zabala-Blanco, and José María Armingol Moreno. "Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review."

[2] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. "ByteTrack: Multi-Object Tracking by Associating Every Detection Box." *Journal Name*.

[3] Yang Gu and Bingfeng Si. "A Novel Lightweight Real-Time Traffic Sign Detection Integration Framework Based on YOLOv4." *Entropy*, vol. 24, no. 4, 2022, p. 487. DOI: 10.3390/e24040487.

[4] Faisal Bashir and Fatih Porikli. "Performance Evaluation of Object Detection and Tracking Systems." Mitsubishi Electric Research Laboratories, Cambridge, MA 02139. Email: fatih@merl.com.