IE643: Deep Learning - Theory and Practice **July-Nov 2024**

Project Report: Speed detection of vehicles from video clip using a single model in different extreme weather conditions

Team Name: Quanta Quotient *Team Member: 22B0656*

Abstract

This project presents the design and development of a Vehicle Speed Detection System utilizing a YOLOv8x model tailored for extreme weather conditions. The model incorporates fine-tuning techniques to enhance its ability to accurately detect and classify vehicles in challenging environments, such as fog, rain, and snow. A robust speed estimation algorithm has been engineered to work seamlessly within the model's pipeline, allowing for precise calculations and displays of vehicle speeds. Additionally, a user-friendly web application has been developed, enabling users to upload video footage for processing. The application provides processed outputs, including bounding boxes and speed estimations, facilitating real-time monitoring and analysis of vehicle speeds under various weather scenarios.

1 Introduction

The rapid advancement of technology and the increasing prevalence of vehicles on the roads have led to a significant need for effective speed detection systems. Accurately measuring vehicle speeds is crucial for traffic management, law enforcement, and enhancing road safety, particularly in adverse weather conditions where traditional detection methods often falter. With the rise of extreme weather events, there is a pressing demand for solutions that can maintain high performance and reliability in diverse scenarios.

This project focuses on developing a Vehicle Speed Detection System using deep learning techniques, specifically the YOLOv8x model. Deep learning has revolutionized the field of computer vision, offering sophisticated methodologies to analyze and interpret visual data. By leveraging deep neural networks, our approach aims to achieve high accuracy in vehicle detection and classification, even in challenging weather conditions such as fog, rain, and snow.

The significance of this project lies in its potential applications, which extend beyond mere speed detection. This technology can enhance traffic law enforcement, inform autonomous vehicle systems, and contribute to the development of intelligent transportation systems that adapt to real-time weather changes.

The structure of this project report is organized as follows:

- We provide a survey of existing literature in Section 3.
- Our proposal for the project is described in Section 4.
- We give details on experiments in Section 6.
- A description of future work is given in Section 8.
- We conclude with a short summary and pointers to forthcoming work in Section 9.

2 Project Workflow

The project workflow outlines the systematic approach taken to address the task of developing a Vehicle Speed Detection System. The following bullet points detail the key stages in the project's progression:

- **Problem Statement:** Clearly defined the problem of accurately detecting vehicle speeds under various weather conditions.
- **Literature Review:** Conducted a comprehensive survey of existing technologies and methodologies related to vehicle speed detection and deep learning applications in computer vision.
- **Model Selection:** Selected YOLOv8x as the primary deep learning architecture due to its efficiency and accuracy in object detection tasks.
- **Data Collection:** Acquired and pre-processed datasets containing various vehicle images under different weather conditions to train the model effectively.
- **Model Training and Fine-Tuning:** Developed and fine-tuned the YOLOv8x model, optimizing parameters for enhanced vehicle detection and classification performance.
- **Speed Estimation Algorithm:** Engineered a speed estimation algorithm that integrates with the detection model, providing real-time speed calculations.
- **Web Application Development:** Built a user-friendly web application that allows users to upload video files and receive processed outputs, including vehicle bounding boxes and speed information.
- **Testing and Validation:** Conducted thorough testing of the model and application to ensure accuracy and robustness across various weather scenarios.
- **Documentation:** Prepared documentation detailing the methodology, results, and insights gained throughout the project.

3 Literature Survey

In the realm of vehicle speed detection and traffic monitoring, several notable studies have emerged, showcasing the application of deep learning techniques to improve accuracy and efficiency. This literature survey outlines key contributions that have informed our project.

3.1 Gollapalli et al. (2023)

Parwateeswar Gollapalli et al. developed a sophisticated system leveraging YOLOv8 for vehicle detection and Deep SORT for multi-object tracking. Their approach focuses on estimating vehicle speeds and counting vehicles in real-time traffic scenarios. The main contributions of their work include:

- **Model Architecture:** The integration of YOLOv8 for robust vehicle detection allowed for rapid and accurate identification of multiple vehicles in a scene.
- **Multi-Object Tracking:** The use of Deep SORT facilitated effective tracking of detected vehicles across frames, essential for accurate speed estimation.
- **Performance Metrics:** Their system demonstrated over 80% accuracy in classifying vehicles and estimating their speeds on real datasets, validating the effectiveness of combining YOLOv8 and Deep SORT for high-speed traffic monitoring.

This work underscores the potential of advanced deep learning models to enhance traffic surveillance capabilities, providing a foundation for further developments in vehicle speed detection.

3.2 Neamah and Karim (2023)

Saif B. Neamah and Abdulamir A. Karim presented a comprehensive real-time traffic monitoring system that integrates YOLOv8 with advanced deep learning techniques. Their study focuses on multiple aspects of traffic analysis, including:

- **Integrated System Components:** The system encompasses vehicle detection, tracking, speed estimation, and size estimation, reflecting a holistic approach to traffic monitoring.
- **Performance Achievements:** Their results indicate high accuracy across various tasks, achieving 96.58% for size estimation, 87.28% for speed estimation, 97.54% for vehicle counting, and 96.58% for detection and tracking. Such metrics highlight the reliability and precision of the proposed system.
- **Data Utilization:** The system was rigorously evaluated using high-resolution video footage (1920x1080) captured at 24 frames per second, which contributed to the accuracy of the model's outputs.

This research not only emphasizes the effectiveness of YOLOv8 in a real-time environment but also showcases the importance of integrating multiple tasks within a single framework for comprehensive traffic monitoring.

3.3 Summary of Existing Approaches

The reviewed literature demonstrates a significant trend toward using advanced deep learning models like YOLOv8 in traffic monitoring applications. The effectiveness of these models in vehicle detection, tracking, and speed estimation across various environments underscores their relevance in addressing the challenges posed by traffic surveillance, particularly under diverse weather conditions.

Our project builds on these foundational works by specifically adapting the YOLOv8 model for extreme weather conditions and developing a user-friendly application to facilitate real-world usage. By leveraging insights from these studies, we aim to enhance the robustness and accuracy of vehicle speed detection in our own implementation.

4 Proposed Approach

This section outlines the proposed approach for vehicle speed detection using deep learning techniques, specifically focusing on the YOLOv8 model. Our methodology aims to enhance the accuracy and reliability of speed estimation, particularly in extreme weather conditions.

4.1 Work Done Before Prep-Presentation Review

Prior to the preparation for the presentation, the team identified the model and approach to be utilized for this project. Key steps included:

- Selection of YOLOv8 as the primary model for vehicle detection, known for its real-time capabilities and accuracy.
- Determination of the speed detection algorithm to be implemented, focusing on effective tracking and measurement techniques across video frames.
- Acquisition of the UA-DETRAC dataset, which provides valuable footage for training and testing the model.

4.2 Work Done After Prep-Presentation Review

Following the prep-presentation review, the team undertook significant steps to refine and implement the proposed approach:

- Coding the logic for vehicle detection and speed estimation, incorporating fine-tuning techniques to enhance model performance in adverse weather conditions.
- Conducting experiments to evaluate the effectiveness of the implemented speed detection algorithm, ensuring robust performance across diverse video inputs.
- Developing a user interface that allows reviewers to upload video clips, with the system generating processed outputs displaying bounding boxes and calculated speeds.

4.3 Neural Network Structure

The architecture of the YOLOv8 model plays a critical role in the success of our project. The key components are as follows:

4.3.1 Backbone

The backbone of YOLOv8 is essential for feature extraction. It processes the input image through a series of convolutional layers, creating a hierarchical feature map that captures various levels of abstraction (from edges and textures to full object parts). This backbone is built upon Convolutional Neural Networks (CNNs), which are fundamental to extracting meaningful features from raw image data.

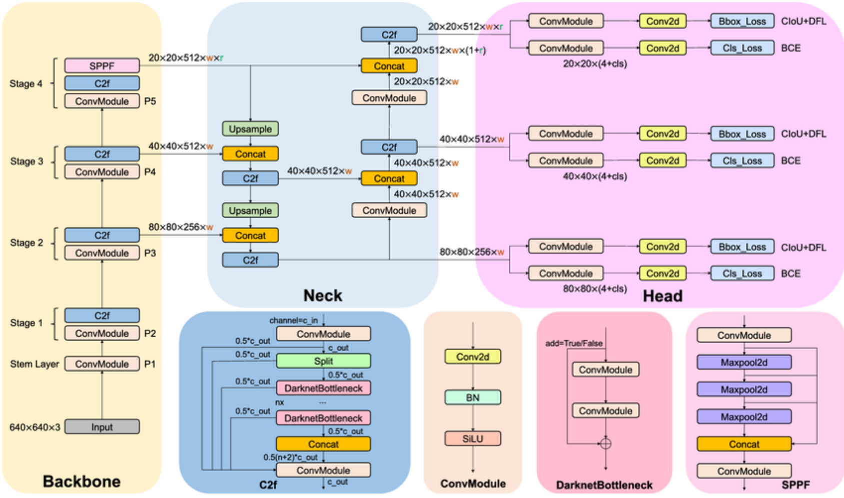

Figure 1: Detailed illustration of YOLOv8 model architecture: The Backbone, Neck, and Head

4.3.2 Neck

The neck of YOLOv8 refines and aggregates the feature maps extracted by the backbone. This process prepares the maps for detection at different scales, thereby improving the model's ability to detect objects of varying sizes. The following components are utilized:

- **Feature Pyramid Networks (FPN):** FPNs combine feature maps from different backbone stages, allowing the model to leverage both high-level and low-level features for enhanced object detection.
- **Path Aggregation Network (PANet):** PANet improves the information flow between feature levels, aiding in accurate object localization by connecting bottom-up and top-down information more effectively.

4.3.3 Head

The head of the model is responsible for actual object detection, where the processed feature maps predict:

- Object bounding boxes
- Class scores
- Confidence scores

Notably, YOLOv8 employs an anchor-free detection approach, moving away from predefined anchor boxes to directly predict bounding boxes relative to a point on the image. This simplification enhances speed and reduces the complexity associated with anchor box tuning. Bounding box predictions are made using CIoU loss, which improves the alignment of predicted boxes.

.

4.4 Speed Estimation Methodology

The vehicle speed estimation process leverages YOLOv8's tracking capabilities across sequential video frames:

- **Frame Processing:** Videos are split into individual frames for YOLOv8 analysis, where each detected vehicle is assigned a unique ID.
- **Distance Calculation:** The distance traveled between consecutive frames is computed, utilizing homography transformation to improve accuracy against perspective and zoom effects.
- **Time Calculation:** The elapsed time is determined based on the number of frames (n) and the video's frame rate (FPS):

$$\text{Total Time} = \frac{n}{\text{FPS}}$$

- **Speed Calculation:** Speed is calculated using the formula:

$$\text{Speed} = \frac{\text{Distance}}{\text{Total Time}} \times \text{Conversion Factor}$$

  This yields speed in desired units (e.g., km/h).

The proposed approach effectively integrates YOLOv8's advanced capabilities with a robust speed estimation algorithm, setting the groundwork for reliable vehicle speed detection across various weather conditions.

5 Dataset Details

This section provides an overview of the datasets used in our vehicle speed detection project, including the UA-DETRAC dataset, its structure, data attributes, preprocessing techniques, and data procurement.

5.1 UA-DETRAC Dataset

The UA-DETRAC (University of Arizona - Detection and Tracking) dataset is a comprehensive collection of video sequences designed for vehicle detection and tracking research. It comprises approximately 100 hours of video footage captured from various locations, offering diverse driving scenarios, lighting conditions, and vehicle types.

5.1.1 Dataset Structure

The UA-DETRAC dataset is organized as follows:

- **Images:**
  - *Training Set:* Contains images in `.jpg` format.
  - *Validation Set:* Also consists of images in `.jpg` format.
- **Labels:**

  - *Training Set:* Contains labels in `.txt` format corresponding to each image.
  - *Validation Set:* Similar structure as the training set.

Each label file has a specific format, where each line corresponds to a detected vehicle and contains the following attributes:

- **Vehicle Class:** An integer representing the type of vehicle (e.g., car, bus, truck).
- **Coordinates:** The center coordinates of the bounding box (x_c, y_c).
- **Dimensions:** The width and height of the bounding box.

The coordinates and dimensions are normalized to the image dimensions for consistency.

5.1.2 Vehicle Classes

The UA-DETRAC dataset includes several vehicle classes, with each class represented by an integer label:

- **Class 0:** Van
- **Class 1:** Car
- **Class 2:** Truck
- **Class 3:** Bus

These classes allow for differentiated detection and tracking capabilities for various vehicle types in the video footage.

5.2 Data Preprocessing Techniques

Given the extensive size of the UA-DETRAC dataset (approximately 100 hours of video), we implemented several preprocessing techniques to optimize the data for our experiments:

- **Frame Selection:** From the original video sequences, one frame was selected for every 15 frames to reduce the dataset size. This resulted in a total of 5,587 images in the training set and 3,756 images in the validation set.
- **Data Augmentation:** To enhance the robustness of our model, each original image was augmented to simulate various weather conditions:
  - Fog
  - Rain
  - Snow

  The augmentation process was carried out using libraries such as OpenCV and Albumentations. As a result, the total dataset size was effectively multiplied by four, yielding additional augmented images and corresponding labels.

5.3 Data Procurement

The UA-DETRAC dataset is publicly available and can be procured from the official website: `http://detrac-db.rit.edu/`. This dataset is widely recognized in the computer vision community for its comprehensive representation of vehicle detection and tracking scenarios.

5.4 Usage in Experiments

The UA-DETRAC dataset will serve as the primary source of data for training and validating our vehicle speed detection model. The preprocessing steps outlined above will ensure that the model is exposed to a variety of vehicle classes and weather conditions, thereby improving its generalization capabilities in real-world scenarios. The augmented images will be used to train the model under different environmental contexts, enhancing its robustness against adverse conditions.

6 Experiments

This section outlines the experiments conducted during the project, detailing the training procedures, optimization algorithms, hardware configuration, and relevant code repositories. Additionally, challenges and alternative approaches considered during experimentation are discussed.

6.1 Training Procedure

The training procedure involved several stages, including data preparation, model fine-tuning, and performance evaluation. We also experimented with different techniques for handling the complexities of vehicle detection and speed estimation in varying weather conditions. Key steps in the training process are outlined below.

6.1.1 Dataset Preparation

Initially, we selected one frame for every 25 frames from the UA-DETRAC dataset. This resulted in a training set that included a wide variety of scenarios but was somewhat sparse. After evaluating the performance, we increased the frame selection rate to every 15 frames, which provided a denser dataset, better capturing vehicle movements. This adjustment enhanced the robustness of our dataset, resulting in a total of 5,587 images for the training set and 3,756 images for the validation set.

In response to weather variability challenges, we augmented the dataset by adding images representing fog, rain, and snow. This augmentation was crucial for improving model robustness, as it simulated conditions that could otherwise degrade detection performance.

6.1.2 Experiments on Pixel to Distance Conversion

One early approach we experimented with involved directly converting pixel measurements to real-world distances (in meters) to estimate vehicle speed. However, a key issue was the inconsistency in pixel-to-meter conversion factors along the road. Due to perspective distortion, objects closer to the camera appeared larger than those farther away, resulting in non-uniform scaling factors along the road. This inconsistency led to inaccurate speed estimations.

To address this, we moved to **perspective transformation**, a method that corrects for the spatial distortion in images taken from a single camera angle. By applying perspective transformation, we achieved more consistent scaling across different areas of the frame, significantly improving the accuracy of distance and speed calculations.

6.1.3 Experiments on Extreme Weather Handling

Given that extreme weather conditions posed challenges to standard object detection, an alternative approach we explored was pre-processing techniques to remove adverse weather effects (e.g., defogging or de-raining). These methods involved image enhancement to reduce the visual impact of extreme weather, allowing the model to detect vehicles more effectively.

While this approach showed some improvement, it was ultimately computationally expensive and could introduce artifacts into images. Therefore, we instead focused on fine-tuning the YOLOv8 model to recognize vehicles directly under these adverse conditions. By training with augmented data simulating various weather scenarios, the model's robustness improved without needing additional pre-processing steps.

6.1.4 Model Fine-Tuning

The model used for vehicle detection and speed estimation is YOLOv8, fine-tuned to adapt to our specific dataset and scenarios. The following settings were applied:

- **Learning Rate:** Initially set to 0.001, and later adjusted based on validation performance. The learning rate was reduced gradually as the model converged to avoid overshooting optimal weights.

- **Batch Size:** A batch size of 16 was used to balance memory constraints and convergence speed. This size allowed for stable training on the available GPU without exhausting memory.

- **Epochs:** Training ran for 50 epochs, with early stopping implemented to prevent overfitting. Early stopping was triggered if validation loss did not improve over five consecutive epochs.

- **Data Augmentation:** In addition to original images, augmented images (e.g., fog, rain, snow) were included to enhance the model's robustness against varying weather conditions. This approach improved detection reliability in real-world weather variations.

6.1.5 Optimization Algorithm

The Stochastic Gradient Descent (SGD) algorithm was used for optimization with the following configurations:

- **Momentum:** Set to 0.9 to enhance gradient descent performance, which accelerated convergence by dampening oscillations.

- **Weight Decay:** A weight decay of 0.0005 was applied to reduce overfitting and encourage generalization by penalizing larger weights.

- **Scheduler:** A learning rate scheduler was implemented, reducing the learning rate by a factor of 0.1 every 10 epochs. This approach helped the model gradually converge without overshooting the optimum, enhancing stability in later training stages.

6.2 Hardware Configuration

The experiments were conducted using a GPU-accelerated environment provided by Kaggle, specifically utilizing the P100 GPU. The hardware configuration is detailed below:

- **GPU:** NVIDIA Tesla P100, a powerful GPU that supports high-throughput operations necessary for deep learning.
- **RAM:** 16 GB of RAM, sufficient to handle the large datasets and high batch sizes used in training.

This setup enabled efficient training of the YOLOv8 model, significantly reducing the time required for both training and evaluation.

6.3 Code and Related Components

The complete code and related components for this project have been uploaded to GitHub. The link to the GitHub repository is provided below:

- **Project GitHub Repository:** `https://github.com/AbhinavPT/DeepLearningSpeedDetecti`

Additional relevant GitHub repositories are referenced below, providing valuable insights into techniques that support this project:

- **YOLOv8 Implementation:** `https://docs.ultralytics.com/models/yolov8/` This repository contains the official YOLOv8 implementation, offering baseline architecture and hyperparameters.

The experimental results from these experiments provided insights into how perspective transformation and model fine-tuning for adverse conditions can lead to enhanced performance in vehicle detection and speed estimation under real-world scenarios.

7 Results

7.1 Performance Metrics

The following table summarizes the performance metrics for the fine-tuned YOLOv8x model on the validation dataset. The results reflect the effectiveness of fine-tuning for object detection in various vehicle categories under different weather conditions.

7.2 Metric Descriptions and Analysis

- **Mean Average Precision (mAP@50)**: This metric represents the model's average precision at an Intersection over Union (IoU) threshold of 50%. The fine-tuned model achieved an mAP@50 of 0.630, indicating a high accuracy in detecting and classifying objects with reasonably overlapping bounding boxes.

Metric	Fine-Tuned YOLOv8x
Mean Average Precision (mAP@50)	0.630
Mean Average Precision (mAP@50-95)	0.475
Precision	0.684
Recall	0.600

Table 1: Performance metrics for fine-tuned YOLOv8x model

- **Mean Average Precision (mAP@50-95)**: Calculated over IoU thresholds from 50% to 95% in steps of 5%, this metric assesses the model's performance with stricter accuracy requirements. An mAP@50-95 of 0.475 shows that the model maintains consistent detection capability across various IoU thresholds.
- **Precision**: Precision quantifies the ratio of correctly predicted objects to all objects predicted by the model. A precision of 0.684 reflects a strong ability to minimize false positives, signifying that the model reliably identifies vehicles as true positives.
- **Recall**: Recall indicates the ratio of correctly predicted objects to the actual objects present in the dataset. With a recall of 0.600, the model demonstrates a good ability to detect the majority of relevant objects, though a slightly higher recall could further reduce missed detections.

7.3 Class-Wise Performance

Table 2 shows the mean Average Precision at 50% IoU (mAP@50) for each vehicle class in the validation set. This breakdown provides insight into the model's accuracy across different object types.

Class	mAP@50
Bus	0.819
Car	0.817
Truck	0.592
Van	0.290

Table 2: Class-wise mAP@50 for fine-tuned YOLOv8x model

7.4 Interpretation of Results

- **Bus and Car Classes**: The model performs exceptionally well in detecting buses and cars, achieving mAP@50 values of 0.819 and 0.817, respectively. This high accuracy indicates that the model is well-suited for applications focused on recognizing these common vehicle types.
- **Truck Class**: The model achieved a moderate accuracy of 0.592 for vans, suggesting room for improvement, potentially by including more varied examples of vans in the training set.
- **Van Class**: With an mAP@50 of 0.290 for the 'others' class, the model struggles to accurately detect less common or irregular vehicle types. This outcome may

reflect lower representation in the dataset, pointing to a need for dataset balancing or additional fine-tuning.

Overall, the fine-tuned YOLOv8x model demonstrates strong performance in detecting mainstream vehicle types like buses and cars, though further tuning may improve detection for less common classes.

8 Plan for Novelty Assessment

The novelty of this project lies in the automation of traffic speed assessment processes that are typically conducted using traditional methods, such as radar guns and manual data collection. In traffic engineering, key metrics such as design speed (the speed at which 98% of vehicles are expected to travel), safe speed (85% of vehicles), median speed (50%), and minimum speed (15%) are determined based on extensive data collection. This data is traditionally gathered from traffic studies using radar guns or similar devices, which can be time-consuming and resource-intensive.

The proposed approach leverages the developed vehicle speed detection model to automate this assessment process. By simply inputting a video feed of traffic along a specific road, the model can effectively detect vehicles, estimate their speeds, and compile statistics relevant to the aforementioned metrics. The steps involved in this novel assessment are outlined as follows:

- **Video Input Processing:** The system will accept a video input from a CCTV or similar surveillance camera positioned along the road. The video will serve as the primary data source for speed estimation.
- **Vehicle Detection and Speed Estimation:** Using the YOLOv8 model trained on various weather conditions, the system will detect vehicles in real time and calculate their speeds. This will be done by applying the perspective transformation technique developed during the project to ensure accurate distance measurements.
- **Data Compilation:** The model will collect speed data over a specified duration, categorizing the speeds into design, safe, median, and minimum speed metrics based on standard traffic engineering definitions.
- **Statistical Analysis:** Automated statistical analysis will be conducted to calculate the 95%ile, 85%ile, 50%ile, and 15%ile speeds from the detected vehicle speeds. This will provide traffic engineers with valuable insights into the flow of traffic and road safety.
- **Reporting Interface:** The system will include a reporting interface that generates visual and statistical reports based on the collected speed data. This will facilitate quick decision-making and aid in road safety assessments.

By automating this traditionally manual process, the project aims to enhance the efficiency of traffic speed assessments and improve road safety management. This innovative approach not only saves time and resources but also provides a more comprehensive understanding of traffic patterns, ultimately contributing to safer and more effective traffic engineering practices.

9 Conclusion

In this project, we tackled the problem of vehicle speed estimation and detection under various extreme weather conditions using a fine-tuned YOLOv8x model. The challenge was to develop a robust system capable of detecting different types of vehicles, such as cars, buses, vans, and other types, while maintaining accuracy across diverse weather scenarios, including fog, rain, and snow.

9.1 Methodology

Our approach involved utilizing the UA-DETRAC dataset, which we augmented to simulate adverse weather effects. We employed Stochastic Gradient Descent (SGD) with momentum for optimization and applied a learning rate scheduler to adaptively adjust the learning rate over the training epochs. Using image augmentation libraries like OpenCV and Albumentations, we generated a diverse dataset, enabling our model to learn effectively across weather conditions. By selecting YOLOv8x for fine-tuning, we leveraged a high-performance object detection framework tailored to our specific requirements for real-time vehicle detection.

9.2 Results and Significance

The results of the fine-tuned YOLOv8x model demonstrate high accuracy in detecting vehicles, particularly in categories such as cars and buses, with mean Average Precision (mAP@50) values of 0.817 and 0.819, respectively. The model also achieved satisfactory precision and recall metrics, reflecting its reliability in minimizing false positives and capturing most relevant objects. However, the lower accuracy for less frequent vehicle types suggests that future work could benefit from additional dataset balancing or further tuning.

This project's findings underscore the potential of deep learning models like YOLOv8x in creating resilient vehicle detection systems for varied and challenging conditions. By adapting this model with custom augmentations and tuning, we have established a framework that could serve as a foundation for advanced traffic monitoring and safety applications. Future work may explore incorporating even more sophisticated weather effects or expanding the model's generalization to other rare vehicle types.

In conclusion, the fine-tuned YOLOv8x model represents a promising solution for real-world scenarios requiring accurate and reliable vehicle detection under diverse weather conditions, with implications for enhancing safety and efficiency in traffic management.

10 References

1. Shaqib S.M., Alo A.P., Ramit S.S., Rupak A.U.H., Khan S.S., & Rahman M.S. (2024). Vehicle Speed Detection System Utilizing YOLOv8: Enhancing Road Safety and Traffic Management for Metropolitan Areas. *arXiv.* `https://arxiv.org/html/2406.07710v1`

2. JuanPedro. (2024). Detailed Explanation of YOLOv8 Architecture. *Medium.* `https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-p`

3. Parwateeswar Gollapalli. (2023). Speed Sense: Smart Traffic Analysis with Deep Learning and Machine Learning. *ResearchGate*.
`https://www.researchgate.net/publication/379497600_Speed_sense_Smart_traffic_analysis_with_deep_learning_and_machine_learning`

4. Saif B. Neamah and Abdulamir A. Karim. (2023). Real-time Traffic Monitoring System based on Deep Learning and YOLOv8. *ARO: The Scientific Journal of Koya University*.
`https://aro.koyauniversity.org/index.php/aro/article/view/1327/35`