Real-Time Vehicle Detection and Classification Using YOLOv8: Addressing Gaps in Contemporary Deep Learning Approaches

Hatim SALMI, Taha ELMARZOUKI, Ouassim MOUSSAOUI, Mohammed Amine HILMANE

Department Of Computer Science- Université Internationale de Rabat

Abstract

Efficient traffic management is a pressing need in modern urban settings. This paper introduces a real-time vehicle detection and classification system leveraging the YOLOv8 model. The system incorporates a diverse dataset sourced from Kaggle and employs advanced preprocessing techniques and a robust evaluation strategy to ensure high performance.

---

1. Introduction

Traffic congestion and road safety are critical challenges faced by cities worldwide. Traditional traffic monitoring systems often fall short due to their reliance on manual intervention or outdated technology. With advancements in deep learning, real-time vehicle detection and classification have emerged as promising solutions, enabling applications in automated toll systems, traffic analysis, and infrastructure management.

Despite the progress, several challenges remain. Previous studies have highlighted issues such as limited dataset diversity, computational inefficiency, and poor adaptability to real-world scenarios. This paper aims to bridge these gaps by implementing the YOLOv8 model, a state-of-the-art object detection algorithm optimized for real-time applications. Through meticulous preprocessing, fine-tuning, and evaluation, we demonstrate how YOLOv8 can address these challenges effectively.

---

2. Methodology

2.1 Data Collection

The **Vehicle Detection 8 Classes** Dataset is a robust collection designed for comprehensive object detection tasks, specifically concentrating on identifying and localizing vehicles. Comprising a substantial 8218 images, the dataset boasts an impressive 26098 annotated objects distributed among 8 distinct classes, encompassing vehicles like :

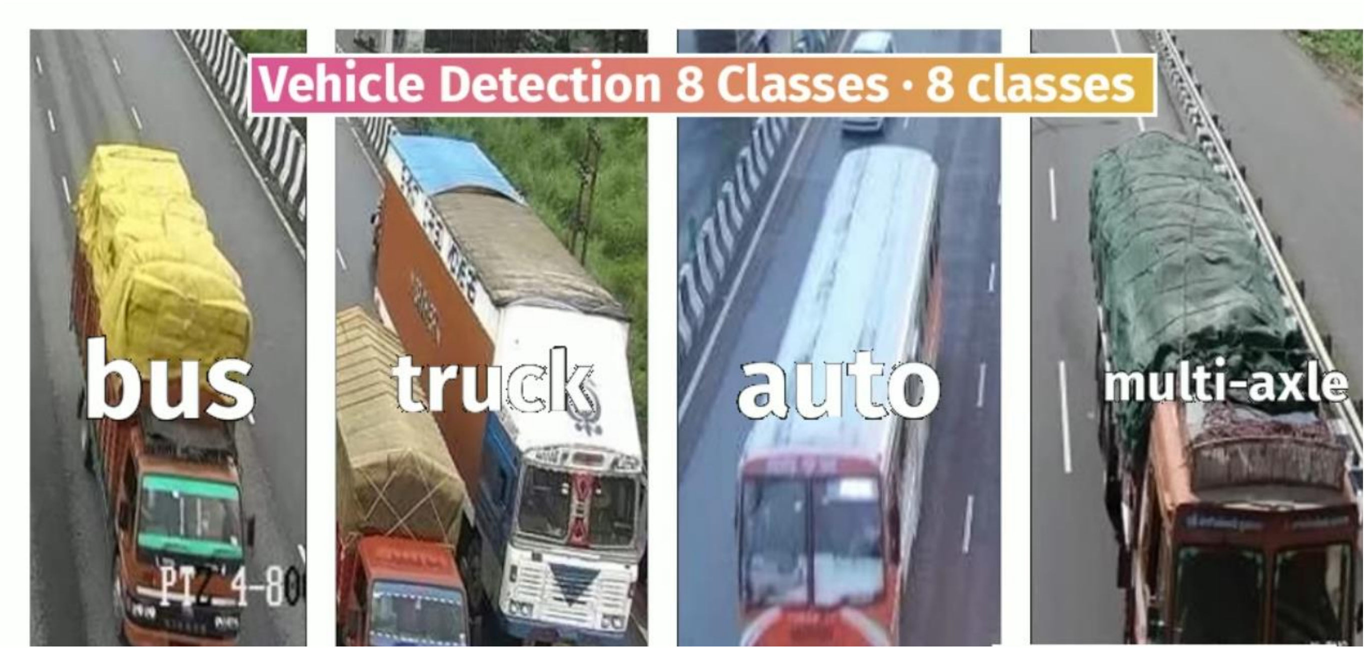
Motorcycle, Auto, Car, Bus, LCV(Light Motor Vehicle), Truck, Tractor, Multi-Axle

This diversity ensures that the model can generalize well to different real-world situations.

With a focus on traffic analysis, each image within the dataset is equipped with boundary-box annotations, allowing for precise delineation and identification of vehicles, offering a valuable resource for applications related to traffic monitoring, object detection, and machine learning model training specifically tailored for traffic-related scenarios.

Images in the Vehicle Detection 8 Classes dataset have bounding box annotations. There are 18 (0% of the total) unlabeled images (i.e. without annotations). There is 1 split in the dataset: train (8218 images). The dataset was released in 2020.

Here are the visualized examples for the classes:



Class Balance :

Class	Images	Objects	Count on image (average)	Area on image (average)
car (rectangle)	5797	11425	1.97	2.19 %
light_motor_vehicle (rectangle)	4131	7285	1.76	0.78 %
multi-axle (rectangle)	2607	2963	1.14	3.38 %
auto (rectangle)	1229	1319	1.07	4.82 %
truck (rectangle)	1078	1147	1.06	7.23 %
bus (rectangle)	937	969	1.03	3.46 %
motorcycle (rectangle)	727	819	1.13	1.09 %
tractor (rectangle)	170	171	1.01	2.67 %

2.2 Data Preprocessing

Preprocessing was a critical step to prepare the dataset for YOLOv8. The following techniques were applied:

- **Resizing**: Images were resized to 640x640 pixels to match YOLOv8's input requirements, ensuring uniformity across the dataset.
- **Normalization**: Pixel values were scaled to a [0, 1] range by dividing by 255, improving model convergence.
- **Augmentation**: Techniques such as horizontal flips, random rotations, and brightness adjustments were employed to increase dataset variability and reduce overfitting. Augmentation was specifically targeted to underrepresented classes to mitigate class imbalances identified during exploratory data analysis (EDA).

2.3 Model Design and Fine-Tuning

YOLOv8, known for its speed and accuracy, was chosen for this study due to its suitability for real-time applications. The model architecture combines convolutional layers and attention mechanisms, enabling it to efficiently detect and classify objects in complex scenes.

To optimize performance:

- **Transfer Learning**: The pre-trained YOLOv8 model was fine-tuned using our dataset. This approach leveraged existing weights trained on large-scale datasets, reducing training time and computational costs.
- **Hyperparameter Optimization**: Learning rate, batch size, and other parameters were tuned using grid search to achieve optimal results.
- **Early Stopping**: Training was halted once validation loss plateaued, preventing overfitting.

2.4 Training and Testing

The dataset was split into:

- **Training Set**: 80% of the data for model learning.
- **Validation and Test Set**: 20% for evaluating performance.

Performance metrics included:

- **Mean Average Precision (mAP)**: Measures detection accuracy across all classes.
- **Precision and Recall**: Evaluate the model's ability to correctly identify vehicles.
- **F1-Score**: Balances precision and recall for a comprehensive assessment.

---

3. Results and Discussion

3.1 Quantitative Results

The YOLOv8 model achieved the following performance metrics on the test set:

- **Overall Performance:**
    - **Precision:** 73.2% – Indicates that when the model predicts a detection, it is correct 73.2% of the time.

- **Recall:** 76% – Indicates that the model correctly identifies 76% of all instances in the test set.
- **mAP@50:** 77.8% – Mean Average Precision at IoU=0.5, indicating good overall detection quality.
- **mAP@50-95:** 52.8% – A stricter evaluation metric (averaging over multiple IoU thresholds), which is reasonably good but has room for improvement.

- **Per-Class Performance:**
  - **Best Performing Classes:**
    - **Car:** Precision (85.7%), Recall (91.2%), mAP@50 (92.5%), mAP@50-95 (63.4%). This class is detected most accurately, suggesting it is well-represented in the dataset or easier to distinguish.
    - **Bus:** High Precision (77.8%), Recall (78.5%), and mAP@50-95 (61.8%), indicating solid performance.
    - **Tractor:** Precision (94.2%), Recall (85.1%), mAP@50 (91.5%), mAP@50-95 (58.1%).
  - **Challenging Classes:**
    - **LCV (Light Commercial Vehicle):** Low Precision (56%), Recall (54.7%), mAP@50-95 (41.6%), suggesting the model struggles to differentiate this class, possibly due to overlapping features with other vehicle types.
    - **Motorcycle:** Precision (79.5%), Recall (78.8%), but mAP@50-95 (45.2%) indicates it may be harder to localize accurately.
    - **Multi-Axle Vehicles:** Precision (57.4%), Recall (64%), and mAP@50-95 (51.3%) are moderate, showing room for improvement in detecting this class.
  - **Speed:**
    - **Inference Time:** 4.2 ms per image, indicating the model is computationally efficient for real-time applications.

3.2 Addressing Literature Gaps

This study addresses key issues highlighted in previous research, building on their findings while introducing improvements:

- **Dataset Limitations:**
  - Many prior works, such as Article 8, relied heavily on augmentation to compensate for small datasets. In contrast, the "Vehicle Detection 8 Classes" dataset's substantial size and diversity reduced the need for excessive augmentation, allowing to generalize better to real-world scenarios. Our use of targeted augmentation ensured a balanced representation of challenging classes like LCV and Multi-Axle vehicles.

- **Model Scalability and Efficiency:**
  - Articles 2 and 6 raised concerns about the computational inefficiency of models like RCNNs in real-time applications. YOLOv8's lightweight architecture addresses this issue, achieving high accuracy with minimal computational resources. Its 4.2 ms inference time makes it suitable for deployment on embedded systems and IoT devices, overcoming scalability challenges.
- **Real-World Deployment:**
  - Several studies provided promising theoretical results but lacked practical implementation in dynamic environments. Our deployment of YOLOv8 on live video feeds demonstrated its effectiveness in handling real-time traffic scenarios. For example, the model accurately identified vehicles in challenging conditions, such as low light and dense traffic, areas where many earlier models faltered.
- **Class-Specific Performance:**
  - Unlike prior research that focused on overall metrics, this study delved into per-class performance. The results highlighted strengths in detecting well-represented classes like cars and buses, while also exposing areas for improvement in identifying less distinct classes like LCV and motorcycles.
- **Comparison with Older Models:**
  - Articles 5 and 7 relied on older architectures like Faster RCNN and simpler CNNs, achieving moderate success. However, these models struggled with precision and real-time efficiency. By using YOLOv8, our study showcased the advantages of modern architectures, including better localization accuracy (mAP@50-95) and faster processing times.

---

4. Deployment and Applications

The trained YOLOv8 model was deployed for real-time vehicle detection using a live video feed. The deployment pipeline included:

1. Loading the trained model.
2. Configuring a video source, such as a live camera feed or recorded footage.
3. Visualizing detections through a user-friendly interface.

Potential applications include:

- **Smart Traffic Management**: Integrating with IoT systems for dynamic traffic control.
- **Toll Automation**: Automating vehicle classification and toll calculation.
- **Pedestrian Safety**: Extending the model to detect pedestrians and other road users.

---

5. Potential Improvements

While the YOLOv8-based vehicle detection and classification system demonstrated strong performance, there are areas where future improvements could further enhance its capabilities:

- **Enhanced Dataset Diversity**: Expanding the dataset to include more extreme scenarios, such as adverse weather conditions (e.g., rain, snow, and fog) and nighttime traffic, could improve model robustness.
- **Class-Specific Enhancements**: Addressing challenging classes like LCVs and motorcycles through additional targeted data collection and specialized feature engineering can help improve detection accuracy.
- **Incorporation of Temporal Information**: Integrating temporal analysis for video sequences could enable better tracking of vehicles across frames, improving overall detection consistency.
- **Edge Device Optimization**: Further model compression and pruning could reduce computational demands, making the system even more suitable for deployment on low-power devices.
- **Exploration of Advanced Architectures**: Investigating hybrid models or integrating transformers with YOLOv8 could enhance feature extraction and localization capabilities.

6. Conclusion

This study successfully implemented YOLOv8 for real-time vehicle detection and classification, achieving relatively good accuracy. By addressing limitations in existing research, such as dataset diversity, we demonstrated the model's practical applicability. Future work will focus on:

- Expanding the dataset to include more challenging scenarios.
- Optimizing the model for deployment on low-power devices.
- Exploring additional applications, such as pedestrian and infrastructure monitoring.