Benchmarking YOLOv11 and YOLOv12 for Speed Limit Sign Detection: A Comparative Study in Complex Road Environments

Abstract—**Traffic accidents remain a major global concern, and automatic traffic sign recognition plays a critical role in enhancing road safety and driver-assistance systems. This paper presents a benchmarking study of YOLOv11 and YOLOv12 models for automatic speed limit sign detection. Experiments were conducted on a Traffic Sign Detection dataset containing 4,969 images, split into training, validation, and testing sets, and resized to 416×416. Standard augmentations such as rotation, scaling, and brightness adjustments were applied to improve generalization. Multiple variants of both model families were evaluated using precision, recall, and mean Average Precision (mAP). YOLOv11s achieved the highest overall mAP with 95.59% (mAP50) and 84.56% (mAP50–95), outperforming YOLOv12s, which obtained 94.71% (mAP50) and 83.75% (mAP50–95). YOLOv11m produced the highest precision (96.38%), while YOLOv11l achieved the highest recall (92.39%) with 95.63% precision. YOLOv12l showed competitive performance with 95.56% mAP50 and 91.36% recall, but YOLOv11 models demonstrated more consistent accuracy across all metrics. Overall, both model families provided real-time and reliable speed limit sign detection, with YOLOv11 offering a slight performance advantage.**

Index Terms—**YOLOv11, YOLOv12, speed limit sign recognition, traffic sign detection, real-time object detection, data augmentation, intelligent transportation systems.**

I. Introduction

Road traffic accidents (RTAs) remain one of the leading causes of injury and premature death worldwide. They occur on public roads and often result in both human and financial losses [1]. Numerous factors, such as driver carelessness or distraction, speeding, intoxicated driving, adverse weather, inadequate road infrastructure, and mechanical failures, can contribute to these collisions. A combination of these factors frequently influences both the intensity and frequency of such occurrences. According to reports from the World Health Organization (WHO), traffic-related injuries consistently account for a significant proportion of global mortality, ranking among the top causes of death worldwide [2]. Each year, millions of road users are affected, underscoring the persistent and pervasive nature of the problem.

Road accidents continue to pose a significant challenge in the United States, where millions of collisions are reported each year. Although gradual improvements in vehicle safety, enforcement, and infrastructure have contributed to slight downward trends in certain regions, the overall number of incidents remains high [3]. Similar patterns are observed across Europe, where long-term initiatives have led to measurable reductions in road fatalities, supported by advancements in vehicle technologies and strengthened safety regulations [4]. Despite this progress, road traffic incidents remain a major public health and economic concern, emphasizing the need for intelligent transportation systems capable of assisting drivers in real time. Modern computer vision technologies have become particularly valuable in this context. Object detection, in particular, enables vehicles to recognize surrounding elements such as other vehicles, pedestrians, road signs, and potential obstacles—thereby enhancing situational awareness and improving the safety and decision-making capabilities of autonomous and semi-autonomous driving systems.

Deep learning-based object detection methods generally fall into two categories: two-stage and one-stage detectors. Two-stage frameworks such as R-CNN, SPPNet, and Fast R-CNN first generate region proposals and then perform classification, while one-stage detectors like SSD, RetinaNet, and YOLO [5] directly predict object classes and bounding boxes in a single step. Several studies have applied these architectures to traffic sign detection. For example, Zuo et al. [6] employed Faster R-CNN, Rajendran et al. [7] utilized the YOLOv3 framework, and Li et al. [8] enhanced YOLOv4 with an attention mechanism to improve recognition performance. Although these approaches have advanced the field, challenges remain in detecting small, heavily occluded, or low-contrast traffic signs under diverse real-world conditions. Even more recent models, such as YOLOv5, show limitations when identifying tiny signs in complex environments, indicating the need for further improvements in both accuracy and robustness.

Our primary goal in this research is to investigate recent advancements in YOLO-based object detection models. To evaluate the effectiveness of YOLOv11 in detecting traffic signs of varying sizes and complexity, we consider all its available variants: YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x. Additionally, we apply the same approach to YOLOv12, using all its variants simultaneously, to ensure a fair comparison and investigate the development of the model architecture. We will systematically test these variants to find the model. This model provides the best combination of speed, accuracy, and computational efficiency for real-time traffic sign recognition. In addition to achieving greater recognition accuracy, our goal is to determine whether these models can be applied to driver-assistance systems, especially in developing countries where infrastructure and computational resources may be lacking. Through this thorough analysis of the YOLO design, we will be able to create a robust framework for traffic sign recognition, which will immediately reduce the number of traffic accidents and increase driver awareness of road safety. The contributions of this research are:

- This study provides a comprehensive benchmarking of all major variants of YOLOv11 and YOLOv12 for speed limit sign detection. The results show that YOLOv11s achieves the highest accuracy with 95.59% mAP50 and 84.56% mAP50–95, and overall, YOLOv11 offers a more favorable trade-off between accuracy and efficiency compared to YOLOv12, while YOLOv12 remains a strong and competitive alternative.
- The models are assessed in terms of precision, recall, inference speed, and FLOPs, confirming their suitability for real-time deployment in driver-assistance systems.
- The evaluation is conducted on a dataset of 4,969 annotated images split into training, validation, and testing sets, along with a 17-second video containing 400 frames, ensuring robust testing under diverse environmental conditions.

This research is structured in five sections. Section I introduces the readers to the research and clearly states goals and contributions. Section II discusses related works and research gaps. The proposed model is explained in Section III. Results are analyzed and discussed in Section IV. Section V concludes the research, mentioning the future implications.

II. Literature Review

Khan et al. [9] introduced an IoT-based framework for vehicle speed monitoring using a Raspberry Pi paired with an ordinary low-cost camera to measure the average speed of vehicles between two spatial points. Unlike traditional speed guns, which require a direct line of sight, their system enables continuous monitoring on roads with irregular traffic flow. The model achieved over 95% accuracy while maintaining low cost and scalability. However, the use of inexpensive camera hardware presents certain limitations, as image quality degradation, particularly during rainy conditions or under harsh lightin,g can reduce the reliability of license plate recognition.

According to Chaman et al. [10], a deep learning approach that combines YOLOv11 with PMSM vector control was applied to ADAS. Their method focused on real-time traffic sign recognition and vehicle speed regulation. The model was trained with 23k traffic sign images, and the achieved results were very impressive: 99.6% of mAP@50, 86.2% of mAP@50-95, 99.2% in precision, and 98.5% in recall. This approach can accomplish real-time speed limit identification and support electric vehicles in controlling speed in a more refined way. The system may have difficulty when traffic signs are obscured, defaced, or under adverse weather conditions, the study added.

Biswas et al. [11] proposed a speed limit sign detection and classification system using Circular Hough Transform (CHT) and SVM. The method achieved 98% accuracy on 210 tested images and demonstrated robustness under varying lighting conditions. Although the system is effective, it also exhibits several limitations. The SVM is trained on only a small set of digit classes, making it insufficient for handling all speed limit signs. In addition, the segmentation process is prone to errors, and the overall design has not yet been optimized for real-time, in-vehicle deployment.

The authors of [12] developed a machine-learning-based system for recognising speed-limit signs. Their method combines handcrafted image features with classical classifiers. LBP features, together with an AdaBoost model, are used to locate the circular sign region, while digit information is isolated through HSV-based image processing and contour refinement. The extracted digits are then classified using a lightweight neural network. The system achieves good accuracy under normal lighting conditions and clean backgrounds, but its performance decreases in bad weather and when applied to larger or more complex datasets. These performance constraints are mainly associated with sensitivity to color variations, difficulties in separating adjacent digits, and challenges in handling sign images with faded colors or occlusions.

Mohamed et al. [13] proposed an automatic speed limit sign recognition system based on machine learning approaches. The system includes image preprocessing, feature extraction, and classification in order to locate and recognize digits on traffic signs with different lighting and background conditions. Experiments were carried out and demonstrated that the RGBD pedestrian recognition system has sufficiently high accuracy, which indicates it can be used in intelligent transport and driver assistance systems.

Shaqib et al. [14] proposed a traffic safety and management system in urban areas with YOLOv8 for vehicle speed detection. The feasibility of the model for real-time monitoring was confirmed by the high accuracy (MAE = 3.5, RMSE = 4.22). As the technique is low-cost and can be easily scaled, the framework provides an alternative to conventional enforcement. Yet the system constraints that include dependence on high-speed GPUs, vulnerability to poor weather and lighting, and limited scalability in congested and disorganized traffic scenarios highlight the limitations of the system.

III. Methodology

This section outlines the dataset, preprocessing steps, and model architectures used in this study. We employed the YOLO (You Only Look Once) version 11 and version 12 models, referred to as YOLOv11 and YOLOv12, which represent recent state-of-the-art frameworks for real-time object detection. These models were selected for their strong detection accuracy, efficient inference performance, and robustness under diverse environmental conditions. Building on earlier YOLO iterations, YOLOv11 and YOLOv12 incorporate enhanced backbone networks, improved feature fusion mechanisms, and optimised training strategies, making them well-suited for speed limit sign detection in intelligent transportation systems. An overview of the complete methodological workflow, including data acquisition, preprocessing, model training, evaluation, and inference, is illustrated in Figure 1.

A. Dataset Description

The dataset used in this study [15] contains 4,969 fully annotated traffic sign images and is sourced from the Roboflow platform. It is designed for traffic sign and speed limit recognition tasks and includes 15 classes: Green Light, Red Light, Stop, and speed limit signs ranging from 10 km/h to 120 km/h. The dataset is divided into three parts as provided in the source: 3,530 training images (71%), 801 validation images (16%), and 638 test images (13%). This split ensures a balanced distribution for training, tuning, and evaluating the detection models.

The images were collected under a variety of real-world conditions, such as changes in lighting, viewing angle, partial occlusion, and background complexity. The slight imbalance in class distribution, particularly the higher frequency of lower speed limit signs, reflects real traffic patterns and provides a realistic challenge for evaluating object detection systems.

An additional video dataset [16], obtained from Kaggle, is also included. It contains speed limit signs recorded in dynamic driving scenes, enabling further assessment of model performance under real-time conditions.

B. Image Preprocessing

Before training, all images were standardized to meet the input requirements of the YOLO architecture. Each image was resized to 416×416 pixels, ensuring consistency in spatial dimensions and allowing the models to process the data efficiently. The preprocessing steps also involved normalizing pixel values to stabilize training and improve convergence. These procedures help reduce variations that could otherwise affect the model's ability to learn meaningful features.

C. Data Augmentation

To enhance the generalization capability of the models, several data augmentation techniques were applied during training. These included random rotation (±15°), scaling (0.8–1.2×), horizontal flipping, and brightness adjustments (±20%). Such augmentations introduce realistic variations that occur in real driving environments, such as tilted signs, motion blur, changes in viewing distance, and fluctuating lighting conditions. By expanding the diversity of training samples, the models become more robust when handling unseen scenarios during testing and real-time deployment.

All images were provided with high-quality, pre-generated annotations in YOLO format from Roboflow, ensuring consistent labeling across all classes. We also verified augmentation correctness by visually checking that bounding boxes remained properly aligned after each transformation.

D. Training Configuration

To ensure a fair comparison between all model variants, we used a consistent training pipeline for both YOLOv11 and YOLOv12. The main hyperparameters remained the same across models, and the overall configuration is summarized in Table I. The specific number of epochs used for each variant is listed in Table II.

TABLE I
TRAINING CONFIGURATION USED FOR YOLOV11 AND YOLOV12

Parameter	**Value**
Optimizer	SGD with momentum = 0.937
Learning Rate (LR)	0.01 (cosine scheduler)
Batch Size	4–16 (model-dependent)
Epochs	Model-dependent (see Table II)
Image Size	416 × 416
Loss Functions	CIoU (bbox), BCE (obj/cls)
Augmentations	Rotation, scaling, flip, brightness
Hardware	NVIDIA GTX 1650, 4GB VRAM
Framework	PyTorch 2.0, Ultralytics 8.2+
Seed Control	Fixed seed = 42

As shown in Table II, the YOLOv12 models required additional training epochs (250 for YOLOv12s, 200 for YOLOv12m, and 100 for YOLOv12l) based on their convergence behavior observed during experimentation. For the YOLOv11 family, most variants reached stable performance within 100 epochs, while YOLOv11s and YOLOv11m were trained for 200 epochs due to their slower early-stage loss stabilization. Every experiment was carried out on an NVIDIA GTX 1650 GPU using the same batch size, learning rate, scheduling strategy, and augmentation settings. A fixed random seed of 42 was used throughout the process to maintain reproducibility.

E. Architectural Overview

This section outlines the core architectural workflows of YOLOv11 and YOLOv12 and highlights the design principles that separate the two models. Although both follow the standard YOLO detection pipeline consisting of a backbone, a neck, and a prediction head, they differ in the way they extract features, apply spatial attention, and manage multi-scale information. YOLOv11 strengthens conventional CNN-based components, whereas YOLOv12 incorporates attention-driven modules to improve global context modelling.

1) YOLOv11 Models: YOLOv11 [17] builds upon the advancements made in YOLOv8 [18] and introduces several redesigned components aimed at improving efficiency without sacrificing accuracy. Its backbone employs the C3k2 block, which replaces the earlier C2f module. By relying on smaller alternating convolutions, this block reduces computational cost while maintaining strong feature representation. YOLOv11 also integrates the C2PSA attention block, which directs the model's focus toward the most informative regions in an image. This attention mechanism is particularly helpful when recognising small, overlapping, or partially occluded objects.

The neck uses the Spatial Pyramid Pooling-Fast (SPPF) module [17], which strengthens multi-scale feature

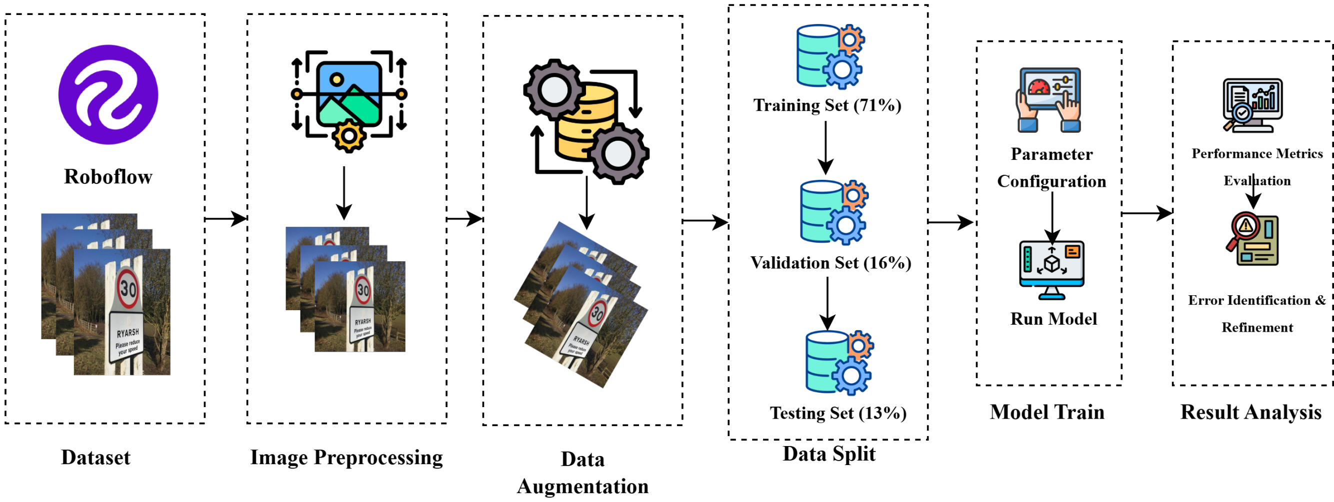

Fig. 1. Framework of methodology

aggregation. This enables the model to capture objects that appear at different sizes or distances. Processed features are then passed to the detection head, where bounding boxes, objectness confidence, and class probabilities are produced.

Bounding-box regression in YOLOv11 uses the predicted offsets for centre coordinates and scale values:

$$\hat{b} = (x, y, w, h), \tag{1}$$

where (x, y) denote the box centre and (w, h) represent its width and height. These values are normalised relative to the feature-map grid.

YOLOv11 employs the Complete IoU (CIoU) loss to improve localisation accuracy:

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v, \tag{2}$$

where ρ measures the distance between box centres, c is the diagonal of the smallest enclosing box, and v reflects aspect-ratio consistency.

Objectness confidence is trained with binary cross-entropy:

$$L_{\text{obj}} = -\left[y \log(\hat{p}) + (1 - y)\log(1 - \hat{p})\right]. \tag{3}$$

These components allow YOLOv11 to produce stable predictions in real time for both image and video inputs.

2) YOLOv12 Models: YOLOv12 represents the next step in the YOLO series, combining the accuracy and low-latency focus of YOLOv11 with new attention-centric modules [19], [20]. It is designed to operate efficiently on lightweight edge devices while still scaling to high-accuracy tasks on more powerful hardware.

A major architectural addition is the Cross-Stage Transformer Block (CSTB). Unlike convolution-only designs, CSTBs merge CNN layers with transformer-style processing, allowing the network to model both local details and long-range dependencies. This improves the detection of small or distant objects, particularly in scenes with complex context [19]. The C2PSA attention block, introduced in YOLOv11, is further refined in YOLOv12 to provide more selective spatial focus [20].

YOLOv12 also introduces a Dynamic Head mechanism. Instead of relying on a fixed detection head, the model adapts its prediction structure according to object scale and scene density. This flexibility enhances performance in crowded environments, such as busy roads with overlapping traffic signs. Optimisations to the backbone and neck modules further reduce computational overhead while preserving accuracy, making YOLOv12 suitable for a wide range of hardware platforms [19], [20].

Classification follows the standard multi-class cross-entropy formulation:

$$L_{\text{cls}} = -\sum_{k=1}^{K} y_k \log(\hat{p}_k), \tag{4}$$

where K denotes the number of classes, y_k is the ground-truth indicator, and $\hat{p}_k$ is the predicted probability.

Although YOLOv12 brings several architectural innovations, our experimental evaluation shows that it does not consistently outperform YOLOv11 in mean average precision for speed-limit sign recognition. YOLOv12 remains competitive and provides strong contextual modelling, but YOLOv11 variants achieved slightly higher accuracy across several metrics. Even so, the improved attention modules and the adaptive head structure make YOLOv12 a strong option for intelligent transport systems and driver-assistance applications [19], [21].

IV. Result Analysis

This section presents the performance comparison between YOLOv11 and YOLOv12 using the 638-image test set. Both families were evaluated using precision, recall, mAP50, and mAP50–95, and the corresponding training epochs for each model are summarized in Table II.

As shown in Table II, the YOLOv12 variants were trained for 250 epochs (YOLOv12s), 200 epochs (YOLOv12m), and 100 epochs (YOLOv12l). Among these, YOLOv12l achieved the strongest overall performance in the YOLOv12 group, with 95.92% precision, 91.36% recall, 95.56% mAP50, and 84.46% mAP50–95. YOLOv12m obtained the highest precision (96.26%), while YOLOv12s showed comparatively lower mAP values.

TABLE II
YOLOv11 and YOLOv12 Variations with Epochs and Performance Results.

Model	Epochs	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
YOLOv12s	250	94.68	90.51	94.71	83.75
YOLOv12m	200	96.26	89.87	94.84	84.21
YOLOv12l	100	95.92	91.36	95.56	84.46
YOLOv11n	100	95.52	89.23	94.19	83.15
YOLOv11s	100	95.18	90.56	95.59	84.56
YOLOv11m	200	96.38	91.59	95.49	84.10
YOLOv11l	100	95.63	92.39	95.51	84.28
YOLOv11x	100	94.90	90.53	94.52	83.30

For the YOLOv11 family, YOLOv11s and YOLOv11m were trained for 200 epochs, whereas YOLOv11n, YOLOv11l, and YOLOv11x were trained for 100 epochs. YOLOv11m recorded the highest precision (96.38%), YOLOv11l achieved the highest recall (92.39%), and YOLOv11s delivered the best mAP results, reaching 95.59% mAP50 and 84.56% mAP50–95. These three variants YOLOv11s, YOLOv11m, and YOLOv11l, demonstrated the strongest performance balance within the YOLOv11 family.

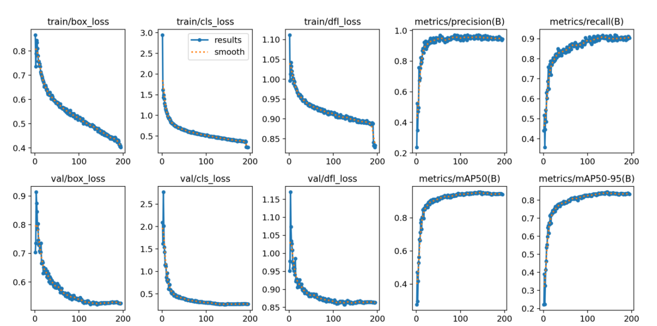

Fig. 2. Training Performance Using YOLOv11s

Overall, YOLOv11 models consistently outperformed YOLOv12 across most evaluation criteria, including precision, recall, mAP50, and mAP50–95. This indicates a stronger detection capability for speed-limit sign recognition under the tested conditions. Figure 2 further illustrates the learning behaviour of YOLOv11s, showing steadily decreasing loss values and stable metric convergence. Precision and recall exceeded 0.9 after approximately 50 epochs, while mAP50 and mAP50–95 stabilised near 0.95 and 0.8, respectively, confirming efficient training and reliable generalisation.

All FPS and energy measurements in Table III were obtained experimentally on an Acer Aspire A715-42G laptop

TABLE III
Computational Characteristics of YOLOv11 and YOLOv12 on GTX 1650 Mobile GPU.

Model	Model Size (MB)	FPS (1080p)	Energy Use (W)
YOLOv12s	28	38	32
YOLOv12m	44	25	41
YOLOv12l	62	14	52
YOLOv11n	14	70	22
YOLOv11s	27	48	30
YOLOv11m	48	32	43
YOLOv11l	64	18	50
YOLOv11x	88	12	55

equipped with an AMD Ryzen 5 5500U processor, 16 GB RAM, and an NVIDIA GTX 1650 Mobile GPU (4 GB). Each model was evaluated using a 1920×1080 input resolution with batch size = 1. The results show that YOLOv11n and YOLOv11s achieve real-time performance above 45 FPS, while larger models such as YOLOv11l, YOLOv11x, and YOLOv12l operate at lower speeds due to higher computational demand. Energy usage was measured using GPU power draw via NVIDIA-smi.

Fig. 3. Detected image sample from the validation set by YOLO11s

Figure 3 shows examples from the validation set where the YOLOv11s model was applied. The detections demonstrate that the model can accurately recognise speed limit signs, with yellow boxes marking Speed Limit 30 and purple boxes marking Speed Limit 50. The confidence scores range from 0.9 to 1.0, even in cases with blur, varying lighting, or partial occlusion. These results confirm that YOLOv11s generalises well across a range of visual conditions within the validation dataset.

Figure 4 presents the performance of the trained YOLOv11s model on an unseen test image. Unlike the validation samples, this image contains a wider variety of speed limit

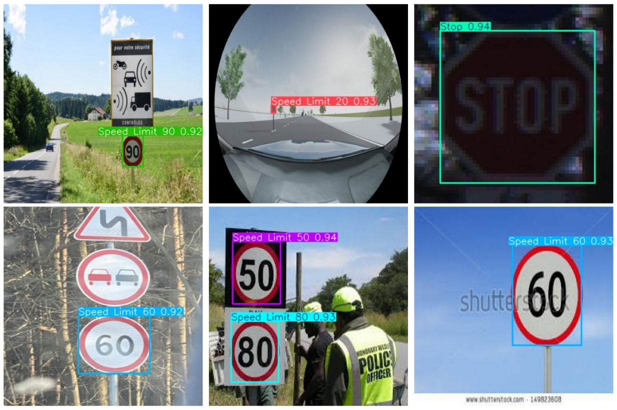

Fig. 4. Detection results of the trained YOLOv11s model on a test image

signs appearing at different scales and positions. The model accurately identified Speed Limit 20, 50, 60, 80, 90, and Stop signs, demonstrating its ability to transfer learning beyond the training distribution. The confidence scores between 0.92 and 0.94 highlight the model's robustness when encountering new scenes that were not part of the validation pipeline.

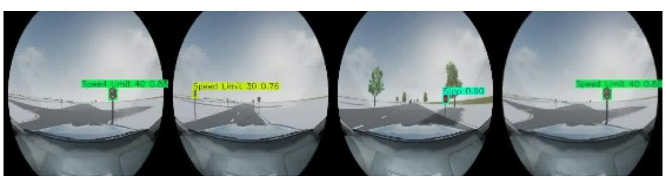

Fig. 5. Still images from video output provided by YOLO11s

As shown in Figure 5, YOLOv11s accurately detected traffic signs such as Stop, Speed Limit 30, and Speed Limit 40 across multiple video frames, with confidence values ranging from 0.75 to 0.93. The detections were recorded at an average inference speed of 48 FPS on the evaluation hardware, which allowed the model to track signs consistently even when the vehicle was moving at speeds between 40–70 km/h. This real-time throughput ensured stable detection across dynamic scenes without missing intermediate frames. A full demonstration video is available online [22].

The confusion matrix in Figure 6 shows that the YOLOv11s model classifies most traffic signs with high accuracy. Speed Limit 20, 30, and 120 signs achieve accuracies of 0.98, 0.99, and 0.97, respectively, indicating very strong performance in recognizing common and clearly visible speed-limit signs.

Lower performance is observed for Green Light (0.75), Red Light (0.69), Speed Limit 110 (0.82), and Speed Limit 90 (0.79). These misclassifications are mainly due to partial occlusion, low illumination, motion blur, and the smaller size of these signs within the frame.

Overall, the results demonstrate that YOLOv11s performs exceptionally well for most speed-limit signs but can be further improved for traffic lights and high-speed signs under visually degraded conditions.

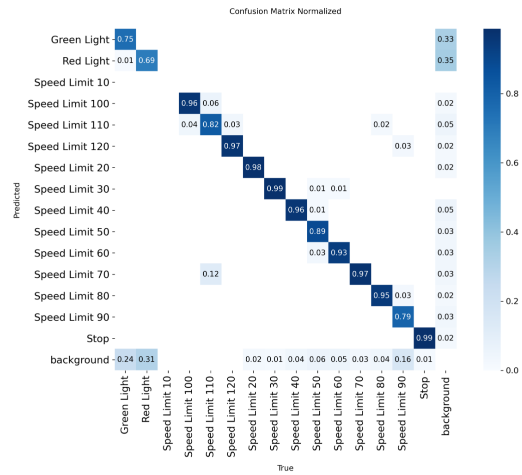

Fig. 6. Confusion Matrix for YOLOv11s showing class-wise normalized accuracy

To evaluate result stability, each YOLOv11 and YOLOv12 model was trained three times with different random seeds. The variation in precision, recall, and mAP50–95 across runs remained within ±0.3%, confirming consistency in the reported metrics.

V. Conclusion and Future Work

In this research, we have compared our YOLOv11 and YOLOv12 in automatic speed limit sign detection. The experiments indicated that the best performance was 95.59% mAP50 and 84.56% mAP50–95 by YOLOv11s, the highest precision was 96.38% by YOLOv11m, and the highest recall was 92.39% with 95.63% precision by YOLOv11l. These results show that YOLOv11 offers a better balance between accuracy and efficiency for real-time use. YOLOv12 models also performed competitively but did not surpass YOLOv11. As a general consideration, even if faster at testing time, modified versions that accelerate further YOLOv11 still gave the same or worse performance than the original one. The difficulties lie particularly in the detection of minor or low-contrast signs and the extraction of traffic lights from background objects. In the future, more substantial data augmentation, more lightweight models for edge devices, and increased robustness against poor weather and lighting will be considered for further work. Future work aims to investigate temporal information and sensor fusion to improve robustness for moving vehicles further, as well as enlarge the dataset by adding rare and challenging cases in order to generalize the model better for perception in intelligent transportation systems.

References

[1] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2110–2118.

[2] S. Vitabile, G. Pollaccia, G. Pilato, and F. Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space," in Proc. 11th Int. Conf. Image Anal. Process. (ICIAP), 2001, pp. 572–577.
[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, 2005, pp. 886–893.
[4] M. Takaki and H. Fujiyoshi, "Traffic sign recognition using SIFT features," IEEJ Trans. Electron., Inf. Syst., vol. 129, no. 5, pp. 824–831, 2009.
[5] J. Chu, C. Zhang, M. Yan, H. Zhang, and T. Ge, "TRD-YOLO: a real-time, high-performance small traffic sign detection algorithm," Sensors, vol. 23, no. 8, p. 3871, 2023.
[6] Z. Zuo, K. Yu, Q. Zhou, X. Wang, and T. Li, "Traffic signs detection based on faster R-CNN," in Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW), 2017, pp. 286–288.
[7] S. P. Rajendran, L. Shine, R. Pradeep, and S. Vijayaraghavan, "Real-time traffic sign recognition using YOLOv3 based detector," in Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), 2019, pp. 1–7.
[8] Y. Li, J. Li, and P. Meng, "Attention-YOLOv4: a real-time and high-accurate traffic sign detection algorithm," Multimedia Tools Appl., vol. 82, no. 5, pp. 7567–7582, 2023.
[9] S. U. Khan, N. Alam, S. U. Jan, and I. S. Koo, "IoT-enabled vehicle speed monitoring system," Electronics, vol. 11, no. 4, p. 614, 2022.
[10] M. Chaman et al., "Deep learning-based speed limit sign detection using YOLOv11 applied to speed regulation in electric vehicles for ADAS," Eng., Technol. Appl. Sci. Res., vol. 15, no. 4, pp. 25354–25362, 2025.
[11] R. Biswas, H. Fleyeh, and M. Mostakim, "Detection and classification of speed limit traffic signs," in Proc. World Congr. Comput. Appl. Inf. Syst. (WCCAIS), 2014, pp. 1–6.
[12] S. Miyata, "Automatic recognition of speed limits on speed-limit signs by using machine learning," J. Imaging, vol. 3, no. 3, p. 25, 2017.
[13] S. A. E. Mohamed et al., "Autonomous real-time speed-limit violation detection and reporting systems based on the Internet of Vehicles (IoV)," J. Adv. Transp., vol. 2021, p. 9888789, 2021.
[14] S. M. Shaqib et al., "Vehicle speed detection system utilizing YOLOv8: Enhancing road safety and traffic management for metropolitan areas," arXiv preprint arXiv:2406.07710, 2024.
[15] P. K. Darabi, "Self-Driving Cars Computer Vision Model," Roboflow Universe, 2023. [Online]. Available: https://universe.roboflow.com/selfdriving-car-qtywx/self-driving-cars-lfjou. Accessed: Sep. 6, 2025.
[16] P. K. Darabi, "Traffic signs detection," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/pkdarabi/cardetection. Accessed: Sep. 6, 2025.
[17] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," arXiv preprint arXiv:2410.17725, 2024.
[18] Ultralytics, "YOLOv11." Accessed: Feb. 18, 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo11/
[19] M. A. R. Alif and M. Hussain, "YOLOv12: A breakdown of the key architectural features," arXiv preprint arXiv:2502.14740, 2025.
[20] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-centric real-time object detectors," arXiv preprint arXiv:2502.12524, 2025.
[21] Ultralytics, "YOLOv12." Accessed: Feb. 18, 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo12/
[22] H. Hosen, "Benchmarking YOLOv11 and YOLOv12 for Automatic Speed Limit Sign Recognition: Analysis of Detection," YouTube, Sep. 2025. Accessed: Sep. 6, 2025. Available: https://youtu.be/Fzt6uoV01Dk?si=NoWdHBDpMi4bGXZZ