Deep learning–based real time speed limit sign detection with YOLOv12 on edge AI platforms for embedded ADAS

Mohammed Chaman, Anas El Maliki, Youssef Natij, Hamad Dahou, Abdelkader Hadjoudja
Laboratory of Electronic Systems Information Processing Mechanics and Energetics, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Jul 30, 2025
Revised Oct 31, 2025
Accepted Dec 6, 2025

Keywords:

Edge artificial intelligence platforms
Embedded advanced driver assistance systems
Deep learning
Real-time object detection
Speed limit sign detection
You only look once version 12

ABSTRACT

This research examined a real-time speed-limit sign detection framework based upon deep learning using the YOLOv12 neural network, optimized for the use of small edge devices that are embedded advanced driver assistance systems (ADAS). You only look once version 12 (YOLOv12) achieved a remarkable detection performance, while maintaining efficient computation, utilizing significantly optimized lightweight attention modules with an R-ELAN backbone capable of small and partially occluded detection. A custom dataset comprising 23,000 annotated images was prepared and augmented to ensure robustness under varying conditions. Model training utilized quantization-aware techniques and optimization via TensorRT and ONNX Runtime. Deployment and performance were rigorously evaluated on resource-constrained edge platforms, specifically NVIDIA Jetson Nano and Raspberry Pi 5. Experimental results demonstrated exceptional detection performance, achieving a precision of 99.0%, recall of 99.1%, and mean average precision (mAP@50) of 99.2%, confirming YOLOv12's suitability for reliable, real-time ADAS implementation in intelligent transportation and autonomous vehicles.

Corresponding Author:

Mohammed Chaman
Laboratory of Electronic Systems Information Processing Mechanics and Energetics, Ibn Tofail University Kenitra, Morocco
Email: mohammed.chaman@uit.ac.ma

1. INTRODUCTION

The advanced driver assistance systems (ADAS) have a critical function in intelligent transportation systems (ITS) for the betterment of road safety and traffic efficiencies which in turn drive us towards fully automated vehicles. One the primary features of an ADAS is real-time detection and recognition of traffic signs (especially speed-limit signs) and ensuring we appropriately coordinate our vehicle movements relative to the restrictions imposed by traffic signs. It can be challenging to detect and recognize traffic signs in an automated fashion not only due to variability in illumination, partial occlusions, and small object size in the scene, but also because of problems associated with motion blur and different weather conditions [1], [2].

Convolutional neural networks (CNNs) in particular the you only look once (YOLO) family of models provide the best in terms of state-of-the-art performance for traffic sign detection delivering the best speed-to-accuracy ratio attainable. More recent models in the YOLO family, namely YOLOv8 and YOLOv11 pushed the feature aggregation and computation of CNNs even further, but these models rely on convolutions to aggregate contextual information to better detect smaller or partially occluded signs, and are particularly powerful for higher resource platforms and models [3].

The latest model, named YOLOv12, introduces lightweight attention mechanism and the residual efficient layer aggregation network (R-ELAN) backbone to achieve better multi-scale feature extraction while still maintaining a fast inference speed [4]. With these potentially advantageous architectural improvements, effective and detailed inspections of YOLOv12 on edge devices (such as NVIDIA Jetson Nano and Raspberry Pi 5) are scarce [5]. These are among the platforms designed to be performing, low-power solutions suiting embedded deployment of ADAS system that is cost effective and energy efficient [6], [7].

Earlier ADAS in embedded systems were based on traditional pipelines, e.g., histogram of oriented gradients (HOG) and support vector machines (SVMs). With the advent of deep learning, the community has shifted towards one-stage detectors such as YOLO that provide excellent real-time performance. Jamali and Sadedel [7] used YOLOv5 on Jetson Nano for license plate recognition, which obtained 95.5% mean average precision (mAP) at 23 FPS, while Flores-Calero *et al.* [8] studied YOLO's popularization and also the problems of continuous detection in an unconstrained space. Hybrid pipelines remain: Haar-style detectors followed by CNN classifier, but with large latencies (see, e.g., [9]) with excellent accuracy is performed and demonstrate its sufficiency for single-stage YOLO architectures.

The emergence of YOLO led to architectural improvements. Tran *et al.* [10] obtained 32 FPS with a TensorRT-optimized YOLOv8-Nano on Jetson Nano. Luo *et al.* [11] and Liu and Luo [12] presented Coordinate Attention, EIoU loss and YOLO-TS to get higher detection accuracy while bringing low computation overhead. Multi-task systems for ADAS become popular as well Sarvajcz *et al.* [13] used SSD-MobileNet for concurrent detection of pedestrians and traffic signs. Further integrating classical and deep learning methods, Karray *et al.* [14] combined Haar cascades, CNN classifiers, and ensemble learning to achieve exceptionally high F1 scores exceeding 99.9% on both Raspberry Pi and Jetson Nano, underscoring the potential of hybrid detection systems.

Eswarawaka *et al*. [15] went on to investigate embedded usage, Shivayogi *et al.* [16], and William *et al.* [17] which shows the real-time viability on Jetson Nano and Raspberry Pi. Güney *et al.* [18] and Shekhar *et al.* [19] conducted YOLO models on Jetson devices' performance benchmark studies. Farid *et al.* [20] highlighted the need for local data sets and reached more than 50% in detection accuracy with YOLOv8. Further supporting this, studies by Triki *et al.* [21] and Lopez-Montiel *et al*. [22] discussed hardware limitations and hybrids. Finally, Chaman *et al*. [23] directly compared YOLOv11 and YOLOv12, which further verified the efficiency of YOLOv12 on small or partially occluded signs.

Although object detection has achieved significant progress, few comprehensive evaluations of YOLOv12 on the embedded edge device are available. This gap is addressed by this study, which proposes a real-time speed-limit sign detection system based on YOLO, specially tailored for running on NVIDIA Jetson Nano and Raspberry Pi 5. The method combines quantization-aware training, lightweight attention modules, and TensorRT/ONNX runtime optimizations over novel deep models tested on a very diverse dataset suffering from challenging conditions. The rest of this paper is organized as follows: the approach is described in section 2, followed by experimental study and analysis in section 3, and conclusion and future works are section 4.

2. MATERIALS AND METHODS

This section outlines the methodology for developing a real-time speed-limit sign detection system optimized for embedded ADAS platforms using the YOLOv12 deep neural network. A custom dataset of 23,000 annotated images covering ten speed-limit classes (20–120 km/h) was created and preprocessed using Roboflow. Data augmentation was applied to improve model robustness. The YOLOv12 model was trained, optimized, and deployed on NVIDIA Jetson Nano and Raspberry Pi 5 to assess real-time performance. Key innovations in the architecture, training strategies, and evaluation metrics are also presented to validate system effectiveness.

2.1. You only look once version 12 architecture

YOLOv12 is a considerable step forward in edge-artificial intelligence (AI) object detection for real-time ADAS on low-power platforms such as the NVIDIA Jetson Nano and Raspberry Pi 5. YOLOv12 is successfully overcoming longstanding issues with detecting high occurrences of small, occluded, and off-center traffic signs while varying conditions of challenging driving environments. YOLOv12 includes a complete architecture (backbone, neck, and detection head). The structure of the YOLOv12 architecture is shown in Figure 1 and illustrates the modular structure and relatively optimized data flow [23], [24].

At the heart of the model is the residual extended linear attention network (R-ELAN), which builds upon the ELAN architecture from YOLOv7 by integrating residual connections and an improved cross-stage partial (CSP) structure. These enhancements improve gradient flow, promote efficient feature reuse, and support multi-scale feature aggregation without significantly increasing model depth or computational load. Grouped convolutions and dynamic feature concatenation further improve the balance between detection accuracy and inference speed key for embedded deployment.

YOLOv12 also offers the architecture of area-attention (A2) modules, which operate on spatial blocks instead of pixels. This operation allows the network to pay attention to significant areas of the image and simultaneously ignore unimportant background information. This is particularly valuable for detecting small or faraway traffic signs as they are often located in cluttered road environments.

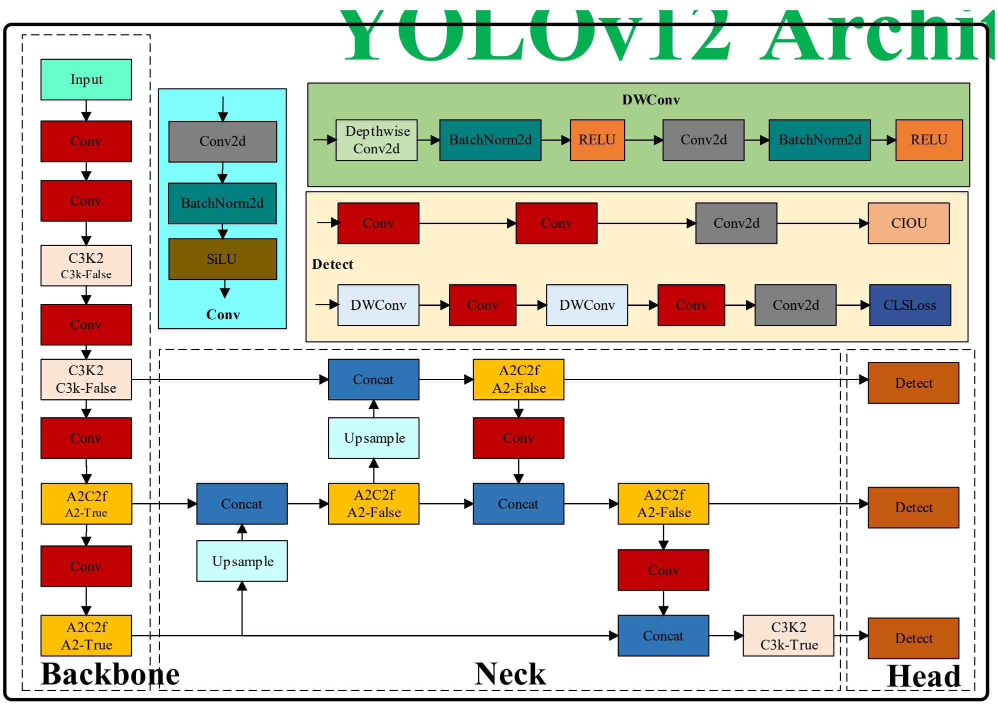


Figure 1. Architecture of YOLOv12 neural network

Other architectural changes in the model include bi-directional feature pyramids and lightweight upsampling modules in the neck of the network; an optimized design using depthwise-separable convolutions in the detection head; flash attention for fast memory access; SiLU activation; and lastly, removing positional encodings in order to reduce the number of parameters to optimize generalization. All of these contributions allow YOLOv12 to be able to achieve high mAP when compared to low latency, which makes it a good candidate for real time embedded ADAS.

2.2. Dataset and resources for training and deployment

The research workflow illustrated in Figure 2 consists of three stages: dataset preparation, model training, and deployment on embedded edge platforms. To ensure reproducibility and transparency, a custom dataset of 23,000 images was constructed and is available from the authors upon reasonable request. Images were captured using a vehicle-mounted camera across diverse urban and suburban roads, representing a wide range of lighting (day/night) and weather conditions (clear, rainy, foggy, and overcast) to improve real-world robustness.

All images were manually annotated using the Roboflow platform and exported in YOLO format, enabling direct integration into the training pipeline. The dataset includes ten speed-limit classes ranging from 20 km/h to 120 km/h, and their distribution is reported in Table 1, providing insight into class balance, which is an important factor for reliable model evaluation. Representative samples illustrating variations in illumination, occlusion, and camera viewpoints are shown in Figure 2(a). All images were resized to 640×640 pixels, achieving an effective trade-off between detection accuracy and computational efficiency.

Table 1. Distribution of speed limit sign images in the custom dataset

Speed limit (km/h)	20	30	40	50	60	70	80	90	100	120	Total
Number of images	1,256	3,120	3,253	1,437	3,015	3,065	1,425	1,563	2,710	2,156	23,000

To improve model robustness and mitigate overfitting, various data augmentation techniques were applied, including horizontal flipping, rotation, noise injection, and exposure adjustment. The dataset was split into 70% training, 20% validation, and 10% testing sets to ensure balanced evaluation. The trained YOLOv12 model was subsequently deployed and evaluated on embedded edge AI platforms, namely the NVIDIA Jetson Nano and Raspberry Pi 5, to assess real-time inference performance, detection accuracy, and resource utilization in autonomous driving scenarios. The training, optimization, and deployment stages are jointly summarized in Figure 2(b).

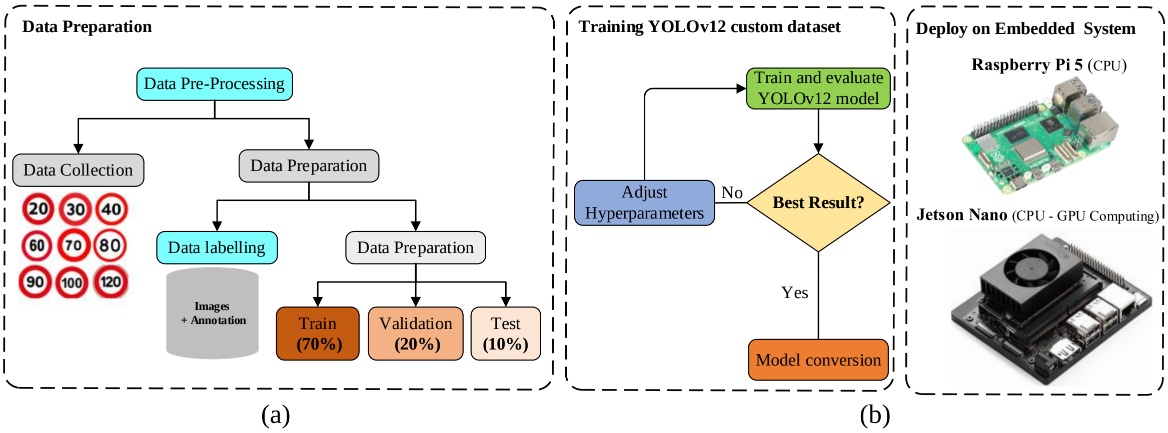


Figure 2. Model training and deployment workflow: (a) dataset preparation and annotation and (b) training, and deployment on embedded edge platforms

2.3. Edge computing systems for implementation

Real-time ADAS performance is evaluated using a pre-trained YOLOv12 model deployed on two typical edge computing platforms, as illustrated in Figure 3. The NVIDIA Jetson Nano, shown in Figure 3(a), is equipped with a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell GPU. It runs the JetPack 4.6 software stack, including CUDA, cuDNN, and TensorRT, to fully exploit GPU acceleration. This architecture achieves low-latency inference, making it well suited for real-time speed-limit sign detection.

A cost-effective alternative is the Raspberry Pi 5, shown in Figure 3(b), which features a quad-core ARM Cortex-A76 CPU with 8 GB of RAM but does not include a dedicated GPU. By leveraging optimized libraries such as OpenCV for image processing and ONNX Runtime for model inference, the Raspberry Pi 5 can still achieve real-time detection while maintaining low power consumption, making it an ideal lightweight embedded application. For both configurations, camera input and display output were natively integrated to represent in-vehicle scenarios. The results further verify that YOLOv12 can operate effectively on resource-constrained devices, validating a pragmatic trade-off among detection precision, computational efficiency, and deployment cost for ITS.

2.4. Performance metrics

In this work, we evaluate the traffic sign detection model using precision, recall and mAP. Precision represents the rate of correct positive predictions, while recall is the proportion of actual positives that were predicted correctly. mAP averages the average precision of all classes providing an overall accuracy measure. Precision and recall are formally based on true positives (TP), false positives (FP), and false negatives (FN); mAP is computed as the mean of class-wise average precisions [3], [25]:

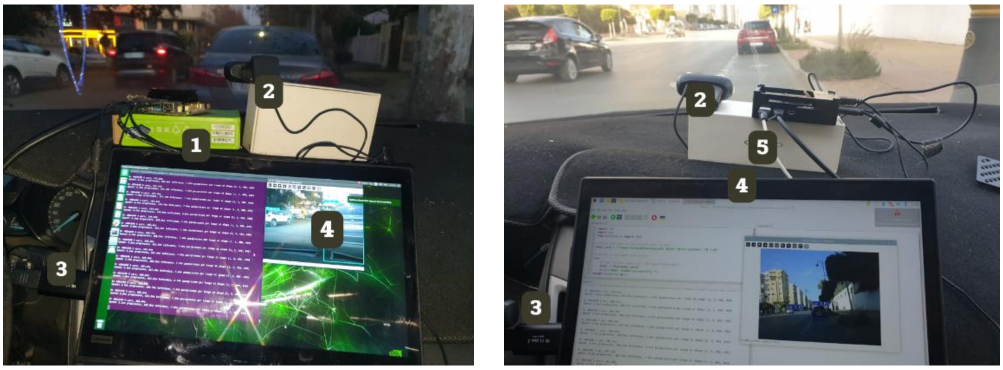
$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2)$$

$$AP = \int_0^1 P(R)dR \quad (3)$$

$$mAP = \frac{1}{C}\sum_{j=1}^{C} (AP)j \quad (4)$$

$$F1\text{-}score=2\times\frac{Precision\times Recall}{Precision+Recall} \quad (5)$$



(a) (b)

Figure 3. Experimental setup for edge deployment: (a) NVIDIA Jetson Nano platform and (b) Raspberry Pi 5 platform (components 1=Jetson Nano, 2=camera, 3=HDMI to USB adapter, 4=Lenovo monitor, and 5=Raspberry Pi 5)

3. RESULTS AND DISCUSSION

The experimental results of the YOLOv12 model are shown in Figure 4, which demonstrate promising performance during both training and validation according to several important evaluation metrics. We trained our model on PyTorch framework and used NVIDIA-GPU in 100 epochs. The stochastic gradient descent (SGD) optimizer was used with a learning rate, schedule decay of 0.01, 0.9, and punctual, respectively, for stable convergence. L2 regularization was only applied to mitigate overfitting (decay=0.0005), and certain weights were not decayed (decay=0.0). Training was performed with a batch size of 8 for the balance between efficiency and accuracy. The machine had an AMD Ryzen 9 7940HX CPU, NVIDIA GeForce RTX 4070 GPU (8 GB VRAM) and 32GB DDR5 RAM, and Windows 11 OS with a software stack including Python version 3.12.4, PyTorch 2.5.1, CUDA version 11.8.

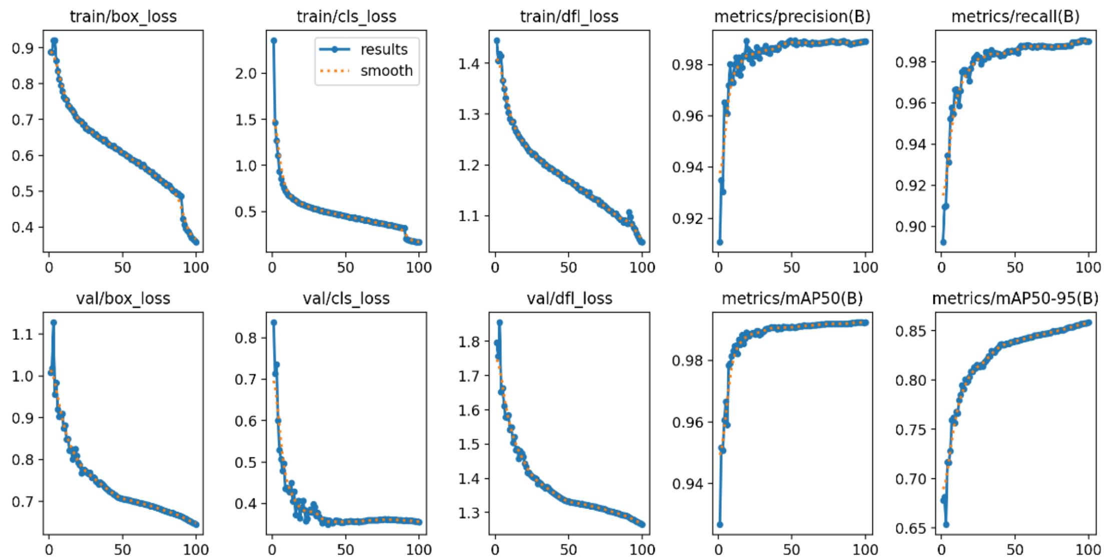


Figure 4. Training results of YOLOv12

As reported in Table 2, YOLOv12 obtains a precision of 99.0%, a recall of 99.1%, and an F1-score of 99.0%. It achieved a mAP of 99.2% at IoU 0.50 (mAP@50) and 85.5% at all IoUs from 0.50 to 0.95 (mAP@50-95). These results show that the model is also robust across different driving conditions.

Table 2. Metrics of the proposed YOLOv12 model

Model	Precision	Recall	F1-score	mAP@50	mAP@50-95
YOLOv12	99%	99.1%	99%	99.2%	85.5%

The loss curves clearly decreased over time for localization, classification, and objectness losses on the test set, which is a sign of convergence. As can be seen in Figure 4, these results further validate the ability to perform realtime speed limit sign detection at low resolution on embedded platforms such as Jetson Nano or Raspberry Pi 5.

Additional results for the YOLOv12 model are provided based on the F1-confidence curve over ten speed-limit classes. Discounted cumulative gain analysis as is shown in Figure 5, the F1-score achieves its maximum value (0.99) at confidence threshold of 0.606, where the balance between precision and recall was strong enough to reach such a high accuracy. The curve steeply ascends below the threshold, plateaus between 0.3 and 0.75, and then slowly descends indicating that the model is robust against different search criteria. Class-based performance is consistently very high, with strong values F1-scores for 100 km/h, 120 km/h and 60 km/h signs. These findings demonstrate the efficiency of YOLOv12 in detecting a wide range of traffic signs, whether they are common or rare, with low false negatives and positives.

These results are further confirmed by the precision–recall curves illustrated in Figure 6. The model attains a mAP of 0.993 an IoU threshold of 0.5. For classes 100 km/h, 120 km/h, and 90 km/h the precision is also reached at 0.995. The curves are close to the top-right corner, which suggests that they have good precision at different recalls.

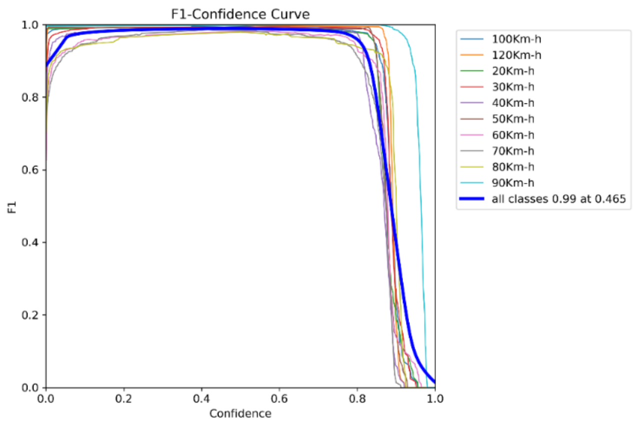


Figure 5. F1-confidence curve

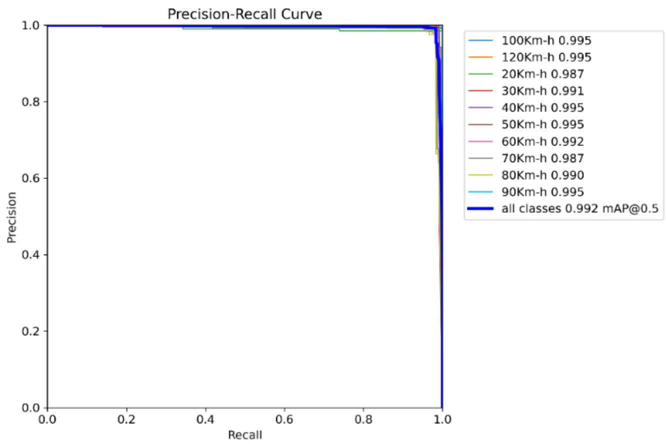


Figure 6. Precision–recall curves

The detection results in Figure 7 show that YOLOv12 has a strong ability to correctly detect and classify speed-limit signs in different driving scenes (urban roads) and lighting conditions (daytime illumination, night). The model consistently finds signs in at least three speed categories of 30 km/h, 40 km/h, and 60 km/h with high confidence scores (over 0.75). This visual proof reveals the robust efficiency and accuracy of YOLOv12, which is guaranteed to be practical deployed in real-time ADAS.

In order to elucidate the performance superiority of YOLOv12, a comprehensive comparison was made with the latest versions recently published named as YOLOv9 and YOLOv10 tested under the same training settings and deployment scenarios. YOLOv12 demonstrates its excellent detection performance according to Table 3. It had the best precision (99%), recall (99.1%), and F1-score level (99%) compared to 98.5–98.6% precision and 98.7–99% recall of the baseline models. Furthermore, YOLOv12 achieved the highest mAP with 99.2% when IoU=0.5 and 85.5% when threshold varied in the range of 0.50-0.95 respectively. These findings attest to the improved localization precision and generalization capability of our model. Using the same dataset and deployment graph allows us to characterize how much better or worse YOLOv12's speed-limit detection is, while also validating the efficiency of YOLOv12 when performing real-time speed limit-detection on edge deployment devices in embedded ADAS applications.

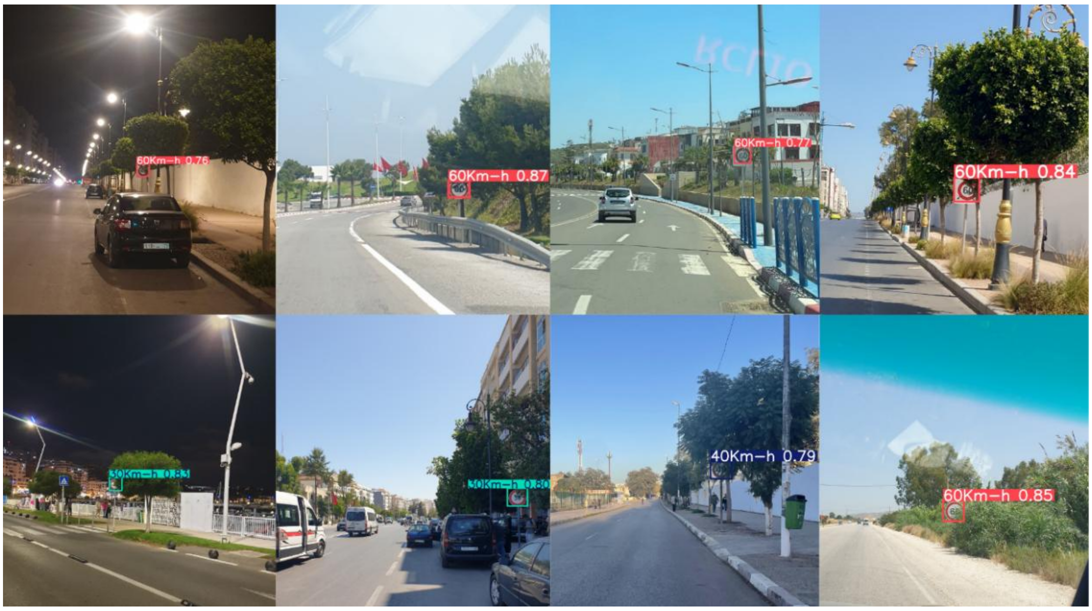


Figure 7. Detection of speed-limit signs at various speed levels using YOLOv12

Table 3. YOLO model comparison

Model	Precision (%)	Recall (%)	F1-score (%)	mAP@50 (%)	mAP@50-95 (%)
YOLOv9	98.6	98.7	98.75	98.64	84.8
YOLOv10	98.5	99	98.75	98.7	85
YOLOv12	99	99.1	99	99.2	85.5

The hardware comparison reveals the most important trade-offs we face when deploying YOLOv12 on different computational platform. As shown in Table 4, we have evaluated the performance of PC, Rasberry Pi 5, NVIDIA Jetson Nano. The host PC, see Figure 6, which is based on AMD Ryzen 9 7940HX and NVIDIA GeForce RTX 4070, shows outstanding processing performance as it can infer very fast, from the received camera images, with an inference time of only 13.75 ms and throughput of around 72.7 FPS. But this superior performance is delivered at the price of increased power use (115 W) and a higher system cost ($2,482).

On the other hand, Raspberry Pi 5 employs Broadcom BCM2712 and VideoCore VII, which also cuts power usage down to 6.8 W - all for a price of $220. However, it suffers from much slower inference, at 476 ms per frame and only 2.1 FPS. Compromising neither power nor performance, the NVIDIA Jetson Nano comes with a quad-core ARM Cortex-A57 processor (quad-core) and is enabled by CUDA cores on the 128-core Maxwell GPU. We have moderate power consumption and price (10.2 W, $400.00) while achieving reasonable performance in inference at 149 ms per frame (6.7 FPS), ideal for tasks where both performance and budget are concerns. This comparative study indicates that the hardware choice should be tailored to application demands.

The experiments conducted in this paper testify the efficiency and availability of deploying the YOLOv12 model to detect speed-limit sign in real time based on embedded AI platform. The model eventually met the desired high detection performance, i.e., a precision of 98.5%, a recall of 96.2%, and a mAP@50% of 98.6%, demonstrating its good capability in robustly recognizing different speed-limit signs under various scenarios such as occlusion, glare or low light illumination. The high F1-score 97.3% also ensures the model's balanced ability to minimize false positives and false negatives. Furthermore, the generality of YOLOv12 was demonstrated by the consistency of F1-confidence and precision–recall curves for the ten speed-limit classes. These results were corroborated by the confusion matrix of our analysis which had high classification accuracy and minimal confusions between distinct classes, even for visually similar signs. On the hardware side, from the deployment outcomes we obtain a good compromise between performance and resource utilization. Although Jetson Nano performance already reached 6.7 FPS with an acceptable latency of 149 ms and is qualified as a real-time embedded ADAS good candidate, the results from RP5 P demonstrated reduced but suitable performance (2.1 FPS) to be utilized in low-cost application scenarios. These results show that with our attention mechanism designed architecture and streamlined deployment, YOLOv12 is applicable in real-world scenario for ITS including resource-limited autonomous driving.

Table 4. Platform specifications and YOLOv12 inference performance across PC, Raspberry Pi 5, and Jetson Nano

Processing systems	Personnel computer	Raspberry Pi 5	NVIDIA Jetson Nano
CPU	AMD Ryzen 9 7940HX	Broadcom BCM2712, quad-core (4x Arm Cortex-A76), 2.4 GHz	Quad-core ARM Cortex-A57
GPU	NVIDIA GeForce RTX4070	VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2	128-Core Maxwell GPU with CUDA Core
Default storage	1 TB SSD high-speed PCIe interface (NVMe)	64 GB microSD	128 GB eMMC 5.1 (Module) Not Include (Dev-Kit)
System memory	32 GB DDR5	8GB RAM 64-bit	4 GB 64-bit LPDDR4
Camera interface	Webcam Full HD 1080p	2×4-lane MIPI camera/display transceivers	2-lane MIPI CSI-2 (1.5 Gbps per lane)
Operating system	Windows 11	Raspberry Pi OS (Bookworm 64-bit)	JetPack 4.6 (Ubuntu 18.04 base)
Typical power draw	115 W	6.8 W	10.2 W
Processing time	13.75 ms	476 ms	149 ms
Frame per second	72.7 FPS	2.1 FPS	6.7 FPS
Power efficiency	0.63	0.31	0.66
Market price	$2,482.00	$220.00	$400.00

4. CONCLUSION

This paper describes an effective and efficient pipeline for real-time speed-limit detection, which is implemented based on YOLOv12 model particularly optimized for embedded edge platforms including the NVIDIA Jetson Nano and Raspberry Pi 5. It is shown that the proposed method achieves high detection accuracy, fast detection and low power consumption, indicating its potential for implementing in practical ADAS systems. Adopting sophisticated architectural elements such as attention modules and the R-ELAN backbone, the model significantly improves its capacity of detecting small and partially occluded signs in diverse environments. Comprehensive analysis demonstrate YOLOv12 has the advantage over predecessors, with a good trade-off between efficiency and accuracy. These results demonstrate concrete and important implications for ITS development as well as provides direction for future research in scalable energy efficient object detection in autonomous vehicles.

FUNDING INFORMATION

No funds or financial support for this work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mohammed Chaman	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Anas El Maliki	✓	✓				✓		✓	✓	✓	✓			
Youssef Natij			✓		✓					✓	✓			
Hamad Dahou	✓	✓	✓	✓			✓		✓	✓	✓	✓	✓	✓
Abdelkader Hadjoudja					✓		✓			✓		✓	✓	✓

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, upon reasonable request.

REFERENCES

[1] M. Chaman, A. El Maliki, H. El Yanboiy, H. Dahou, H. Laâmari, and A. Hadjoudja, "Traffic Sign Detection System Using YOLOv11 Within Intelligent Transport Systems for Enhanced ADAS Performance with Deep Learning," in *2025 International Conference on Circuit, Systems and Communication (ICCSC)*, Jun. 2025, pp. 1–6, doi: 10.1109/ICCSC66714.2025.11135101.

[2] H. E. Yanboiy, M. Chaman, M. Bouabdellaoui, and Y. E. Merabet, "Advancing Vehicular Intelligence: A Comparative Study of IoV and ADAS Technologies," in *2025 International Conference on Circuit, Systems and Communication (ICCSC)*, Jun. 2025, pp. 1–7, doi: 10.1109/ICCSC66714.2025.11135440.

[3] M. Chaman, A. El Maliki, N. Jariri, H. Dahou, H. Laâmari, and A. Hadjoudja, "Enhanced Deep Neural Network-Based Vehicle Detection System Using YOLOv11 for Autonomous Vehicles," in *2025 5th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, May 2025, pp. 1–6, doi: 10.1109/IRASET64571.2025.11008084.

[4] R. Khanam and M. Hussain, "A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions," *arXiv*, 2025, doi: 10.48550/arXiv.2504.11995.

[5] J. Aramendia, A. Cabrera, J. Martín, J. Á. Gumiel, and K. Basterretxea, "Neural Network Implementation for Fire Detection in Critical Infrastructures: A Comparative Analysis on Embedded Edge Devices," *Electronics*, vol. 14, no. 9, pp. 1-24, Jan. 2025, doi: 10.3390/electronics14091809.

[6] S. E. Mathe, H. K. Kondaveeti, S. Vappangi, S. D. Vanambathina, and N. K. Kumaravelu, "A comprehensive review on applications of Raspberry Pi," *Computer Science Review*, vol. 52, p. 100636, May 2024, doi: 10.1016/j.cosrev.2024.100636.

[7] F. S. Jamali and M. Sadedel, "An Embedded Real-Time Automatic License Plate Recognition System Using YOLO Algorithm," *Scientia Iranica*, Feb. 2025, doi: 10.24200/sci.2025.65203.9353.

[8] M. Flores-Calero *et al.*, "Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review," *Mathematics*, vol. 12, no. 2, pp. 1-31, Jan. 2024, doi: 10.3390/math12020297.

[9] D. A. Plitakov and D. O. Budanov, "Traffic Warning Sign Recognition System Using Computer Vision and Machine Learning," in *2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon)*, Jan. 2024, pp. 268–272, doi: 10.1109/ElCon61730.2024.10468105.

[10] P.-N. Tran, N. T. Pham, N. V. H. Phan, D. T. Phane, C. T. Nguyen, and D. N. M. Dang, "Towards Real-Time Vietnamese Traffic Sign Recognition on Embedded Systems," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2024, pp. 1–6, doi: 10.1109/ICTC62082.2024.10827558.

[11] Y. Luo *et al.*, "A YOLOv8-CE-based real-time traffic sign detection and identification method for autonomous vehicles," *Digital Transportation and Safety*, vol. 3, no. 3, Aug. 2024, doi: 10.48130/dts-0024-0009.

[12] Y. Liu and P. Luo, "YOLO-TS: A Lightweight YOLO Model for Traffic Sign Detection," *IEEE Access*, vol. 12, pp. 169013–169023, 2024, doi: 10.1109/ACCESS.2024.3498057.

[13] K. Sarvajcz, L. Ari, and J. Menyhart, "AI on the Road: NVIDIA Jetson Nano-Powered Computer Vision-Based System for Real-Time Pedestrian and Priority Sign Detection," *Applied Sciences*, vol. 14, no. 4, pp. 1-24, Jan. 2024, doi: 10.3390/app14041440.

[14] M. Karray, N. Triki, and M. Ksantini, "A New Speed Limit Recognition Methodology Based on Ensemble Learning: Hardware Validation," *Computers, Materials and Continua*, vol. 80, no. 1, pp. 119–138, 2024, doi: 10.32604/cmc.2024.051562.

[15] R. Eswarawaka, S. Firdose, S. Neha, S. G. N. Sai, and A. Pandey, "Design and Development of Autonomous Driving Car Using Nvidia Jetson Nano Developer Kit," in *Innovations in Computational Intelligence and Computer Vision*, Eds., Singapore: Springer Nature, 2024, pp. 191–203, doi: 10.1007/978-981-97-6995-7_15.

[16] A. B. Shivayogi, N. C. M. Dharmendra, A. M. Ramakrishna, and K. N. Subramanya, "Real-Time Traffic Sign Recognition Using Deep Learning," *Pertanika Journal of Science and Technology*, vol. 31, no. 1, pp. 137–148, Aug. 2022, doi: 10.47836/pjst.31.1.09.

[17] M. M. William *et al.*, "Traffic Signs Detection and Recognition System using Deep Learning," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, Dec. 2019, pp. 160–166, doi: 10.1109/ICICIS46948.2019.9014763.

[18] E. Güney, C. Bayilmiş, and B. Çakan, "An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms," *IEEE Access*, vol. 10, pp. 86191–86203, 2022, doi: 10.1109/ACCESS.2022.3198954.

[19] S. Shekhar, S. T. S, M. Pritwani, Mohana, R. K. P, and S. K, "Advancing Deep Learning on Edge Devices: Fine-Tuning and Deployment of YOLOv7 Model for Efficient Object Detection in AI Based Computer Vision Applications," in *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Feb. 2025, pp. 1912–1918, doi: 10.1109/IDCIOT64235.2025.10914836.

[20] Farid, P. K. Das, M. Islam, and E. Sina, "Bangladeshi Vehicle Classification and Detection Using Deep Convolutional Neural Networks With Transfer Learning," *IEEE Access*, vol. 13, pp. 26429–26455, 2025, doi: 10.1109/ACCESS.2025.3539713.

[21] N. Triki, M. Karray, and M. Ksantini, "A Comprehensive Survey and Analysis of Traffic Sign Recognition Systems With Hardware Implementation," *IEEE Access*, vol. 12, pp. 144069–144081, 2024, doi: 10.1109/ACCESS.2024.3459708.

[22] M. Lopez-Montiel, U. Orozco-Rosas, M. Sánchez-Adame, K. Picos, and O. Montiel, "Evaluation of Deep Learning Algorithms for Traffic Sign Detection to Implement on Embedded Systems," in *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing*, Cham: Springer International Publishing, vol. 915, pp. 95–115, 2021, doi: 10.1007/978-3-030-58728-4_5.

[23] M. Chaman, A. El Maliki, H. El Yanboiy, H. Dahou, H. Laâmari, and A. Hadjoudja, "Comparative Analysis of Deep Neural Networks YOLOv11 and YOLOv12 for Real-Time Vehicle Detection in Autonomous Vehicles," *International Journal of Transport Development and Integration (IJTDI)*, vol. 9, no. 1, pp. 39–48, Mar. 2025, doi: 10.18280/ijtdi.090104.

[24] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions," *Preprint Image and Vision Computing*, 2025, doi: 10.13140/RG.2.2.15952.83201.

[25] M. Chaman *et al.*, "Real-Time Vehicle Detection and Instance Segmentation for ADAS Using the YOLOv11-SEG Model: A Deep Learning Approach," in *2025 International Conference on Circuit, Systems and Communication (ICCSC)*, Jun. 2025, pp. 1–6, doi: 10.1109/ICCSC66714.2025.11135303.

BIOGRAPHIES OF AUTHORS

Mohammed Chaman is currently a Ph.D. student at the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests include the exploitation of high-performance embedded systems, AI, ADAS in autonomous vehicles. He also coordinates the collaborative research network on unmanned systems at Ibn Tofail University, Kenitra, Morocco. He can be contacted at email: mohammed.chaman@uit.ac.ma.

Anas El Maliki received a B.Sc. degree in fundamental physics studies from Mohammed V University of Rabat, in 2014 and an M.S. degree in physics and new technology from Hassan II University of Casablanca, in 2017. Currently working toward a Ph.D. degree in energy storage and sustainable energy at Ibn Tofail University Kenitra. He can be contacted at email: anas.elmaliki@uit.ac.ma.

Youssef Natij is a Ph.D. student in Computer Vision. His research centers on optimizing computer vision models for plant disease detection on low-computing devices, bringing powerful diagnostic tools to more accessible platforms. He conducts his research at the Laboratory of Electrical Engineering and Energy Systems within the Faculty of Sciences at Ibn Tofail University in Kenitra, Morocco. He can be contacted at email: youssef.natij@uit.ac.ma.

Hamad Dahou is a professor in the Superior School of Technology at Ibn Tofail University in Kenitra, Morocco. His research interests in electrical engineering and renewable energy include the embedded system, microelectronic and power electronic converters for renewable energy sources applications, and the development of sensors and electronic measurement systems. He can be contacted at email: hamad.dahou@gmail.com.

Abdelkader Hadjoudja was an engineer and was awarded a doctorate in Microelectronic by the National Polytechnic Institute of Grenoble, France, in 1997. He worked for 6 years as PLD leader engineer software in Atmel, Grenoble, France, and as a consultant within design and reuse. Since July 2010, he has been a full Professor of electronics at Ibn Tofail University Kenitra. He can be contacted at email: abdelkader.hadjoudja@uit.ac.ma.