Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 2, April 2026, pp. 1640~1652
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.11215

Embedded deployment of traffic sign detection and recognition systems

Imane Taouqi[1], Mohamed Lamane[1,2], Abdessamad Klilou[1], Assia Arsalane[1,3], Kebir Chaji[1]
[1]Microelectronics, Embedded Systems, and Telecommunications (MiSET) Laboratory, Department of Electrical Engineering, Faculty of Sciences and Technologies, Sultan Moulay Slimane University, Beni Mellal, Morocco
[2]Multidisciplinary Research and Innovation Laboratory, Moroccan School of Engineering Sciences Casablanca, Casablanca, Morocco
[3] Mechatronics Department, Higher School of Technology, Sultan Moulay Slimane University, Beni Mellal, Morocco

Article Info

Article history:

Received Aug 14, 2025
Revised Feb 5, 2026
Accepted Mar 5, 2026

Keywords:

Advanced driver assistance systems
Computer vision
Detection
Embedded systems
You only look once version 7

ABSTRACT

Traffic sign (TS) detection and recognition are essential components of advanced driver assistance systems (ADAS), contributing to safer and more reliable driving. However, deploying deep learning–based vision models on embedded platforms is challenging due to constraints in computational power and energy consumption. In this work, a comparative deployment of you only look once version 7 (YOLOv7) and YOLOv7-tiny deep learning algorithms is conducted on embedded NVIDIA platforms, namely Jetson Nano and Jetson Xavier NX, to evaluate their suitability for real-time TS detection. Following the detection stage, a convolutional neural network (CNN) is integrated to perform TS recognition, enabling a complete detection–recognition pipeline. Experimental results show that YOLOv7-tiny achieves higher detection precision of 97%, while providing better speed and computational cost on resource-constrained devices, with Jetson Nano reaching 18.8 frames per second (FPS) and, on Jetson Xavier NX reaching 43 FPS. The integrated CNN model ensures reliable classification of detected TS with an accuracy of 99.54%. This work highlights the trade-offs between precision, speed, and power consumption and provides practical guidance for selecting detection and recognition architectures for embedded ADAS applications.

Corresponding Author:

Imane Taouqi
Microelectronics, Embedded Systems, and Telecommunications (MiSET) Laboratory
Department of Electrical Engineering, Faculty of Sciences and Technologies
Sultan Moulay Slimane University
523, Beni Mellal, Morocco
Email : imane.taouqi@usms.ma

1. INTRODUCTION

Traffic sign (TS) detection and recognition are fundamental capabilities in advanced driver assistance systems (ADAS) and autonomous driving, enabling compliance with traffic regulations, improving driver awareness, and enhancing road safety [1]. Our previous work [2] demonstrated the good performances of the you only look once version 7 (YOLOv7) network for TS detection and recognition in Moroccan road scenes, achieving promising results in terms of precision and real time inference. However, despite advances in deep learning based object detection, the transition from laboratory settings to real world autonomous vehicle applications presents significant challenges [3], particularly in terms of deploying these systems on embedded hardware with limited computational resources.

The automotive industry requires efficient, reliable, and cost effective solutions that can operate in real-time within the constrained environment [4] of a vehicle's onboard systems [5]. Even high-performance graphics processing units (GPUs) can deliver impressive results in research settings [6], they are often impractical for deployment in production vehicles due to power consumption [7], space constraints and cost. Consequently, there is a pressing need to adapt and optimize vision models to run efficiently on resource-constrained embedded platforms without compromising detection performance [8].

In recent years, a diverse array of embedded systems have emerged [9], each with unique characteristics suited to different aspects of artificial intelligence (AI) implementation, particularly in terms of frames per second (FPS), which is heavily dependent on the hardware platform used. This variability presents significant challenges when aiming for real-time performance on edge embedded hardware. Then comprehensive evaluation of computer vision algorithms across diverse hardware platforms is essential for real-world deployment. Zagitov *et al*. [10] conducted a comprehensive evaluation of modern object detectors on low power devices for real time object detection, including YOLO different versions, EfficientDet Lite and Mobilenet-single shot multibox detector (SSD). Their work established that for image size 512x512 SSD Mobilenet V2 models generally offer faster inference times compared to the other methods used, achieving 5.6 FPS using Jetson Nano, 4.5 FPS using Raspberry Pi 4B and 0.9 FPS using Raspberry 3. These results can be further improved using recent lightweight detection algorithms. Zhu *et al*. [11], performance evaluations were conducted by implementing YOLOv3 and PaddlePaddle you only look once (PPYOLO) [12] algorithms on the Jetson Nano and Jetson Xavier NX platforms, utilizing two different input resolutions, 320×320 and 608×608. The PPYOLO algorithm demonstrated superior processing speed compared to YOLOv3 across both computing devices. Their findings revealed that input resolution substantially impacts processing throughput, with reduced dimensions yielding accelerated inference rates. Nevertheless, employing smaller input dimensions results in diminished detection precision. Civik and Yuzgec [13] proposed a novel approach using deep learning to detect driver fatigue in real-time on the NVIDIA Jetson Nano, despite optimizations, they achieved only 6 FPS. Sarvajcz *et al*. [14], SSD-MobileNet implementations, typically achieved 8.7 FPS on the NVIDIA Jetson Nano. Furthermore, in Luo *et al*. [15] the deployment of YOLOv8 into Raspberry Pi 4B reaches 0.79 s (1.26 FPS), illustrating the significant challenge of attaining real-time performance on lower-cost hardware. Haijoub *et al*. [16] deployment of YOLOv8 for detection and tracking unmanned surface vehicles on the NVIDIA Jetson TX2 platform, achieved a 99% mean average precision (mAP) and an operational speed of 17.99 FPS, with energy consumption of 5.61 watts. This performance approaches real-time capabilities for many applications. Lopez-Montiel *et al*. [17] achieved 90 FPS on Jetson Nano and 253 FPS on Jetson Xavier AGX. While FPS is high, it's a classifier, not a detection pipeline. However, beyond the deployment aspect, complete detection and recognition pipelines are less frequently benchmarked end-to-end on embedded devices, limiting practical guidance for ADAS deployment.

Building upon our previous research [2], the present paper aims to address these challenges by focusing on the deployment of YOLOv7 based TS detection and recognition system across two distinct embedded platforms Jetson Nano [18] and Jetson Xavier NX from NVIDIA [19]. The Jetson Nano, with its 128-core Maxwell GPU and modest 5-10W power consumption [20], represents an accessible entry point for low cost automotive applications, while the Xavier, featuring a more powerful Volta architecture with dedicated Tensor cores and expanded memory bandwidth, offers enhanced capabilities for automotive applications with power consumption range of 10-20 W [21]. These embedded AI platforms represent different points in the performance cost spectrum for edge computing applications [22] allowing evaluation deployment strategies across varying resource constraints. This work explores the critical balance between detection precision, inference speed and power consumption within these different embedded platforms, which provides practical insights for embedded deployment of TS detection systems in ADAS. In this paper, we present several key contributions:

- Deployment of YOLOv7 and optimized YOLOv7-tiny on the proposed embedded platforms.
- Comprehensive evaluation of the system's performance in terms of FPS.
- Analysis of the model's robustness when deployed on embedded systems, considering factors such as power consumption.
- Expansion of our recognition dataset to forty-three TS classes.
- Integration of recognition model within YOLOv7 detection model to evaluate the performance of the whole system.

By addressing these aspects, our research aims to bridge the gap between state-of-the-art TS detection models and their practical implementation in autonomous vehicles. The insights gained from this study will contribute to the development of more efficient, reliable, and deployable perception systems for next-generation intelligent transportation systems.

The remainder of this paper is organized as follows: section 2 describes method used, including deep learning model overview, embedded system implementation, and experimental setup. Section 3 presents

and discusses the results of our experiments, focusing on the performance and efficiency of the embedded systems used. Finally, section 4 concludes the paper and outlines directions for future research.

2. METHOD

2.1. Deep learning algorithms overview

The YOLOv7 network architecture employs an efficient backbone denominated extended efficient layer aggregation network (E-ELAN) [23], which enhances feature propagation and utilizes computational resources more effectively. Its architecture consists of three main parts. First the backbone extracts essential features from the input image using a combination of convolutional, batch normalization (BN), Leaky rectified linear unit (ReLU) activation function (CBL) layers and multi-scale convolutional blocks (MCB) [24]. Max pooling (MP) layers down sample feature maps, reducing spatial dimensions while retaining important information. The final stage includes a spatial pyramid pooling cross stage partial concatenation (SPPCSPC) module, which captures multi-scale features to enhance the model's ability to detect objects of different sizes. Second, the neck further processes the extracted features to strengthen spatial and semantic information. It uses a combination of CBL layers, MCBs, and concatenation layers to fuse features from different scales. These fused features are refined using upsampling layers, improving the detection of smaller objects. Finally, the head generates the final predictions for object detection. It consists of multiple CBL layers to process the fused features and produce output feature maps at different resolutions (80×80, 40×40, and 20×20), enabling detection across multiple scales. Each output contains object classifications, bounding box coordinates, and confidence scores.

The overall YOLOv7-tiny architecture used in this work is illustrated in Figure 1, while the detailed internal structure of its core building blocks (MCB and SPPCSP) are shown in Figure 2. The tiny variant significantly reduces the computational requirements, making it more suitable for real-time processing on embedded systems.

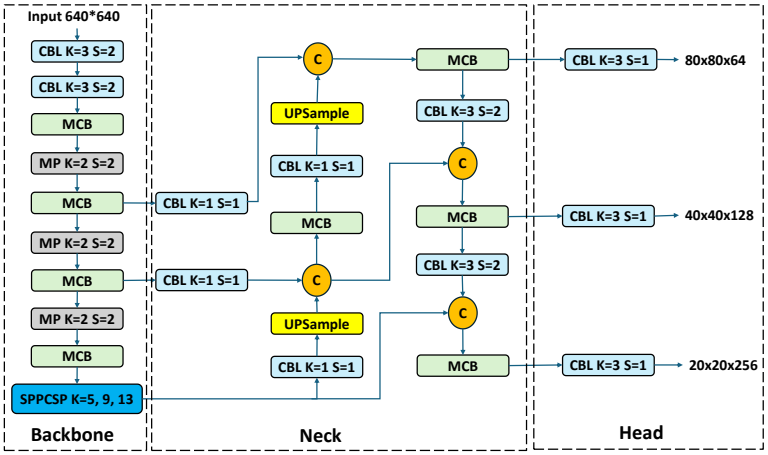


Figure 1. YOLOv7 tiny architecture

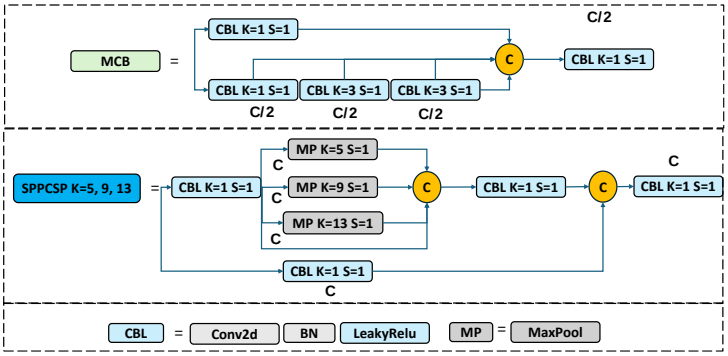


Figure 2. Detailed view of the YOLOv7-tiny modules

In the other hand, a proposed CNN sequential architecture, presented in Figure 3, was used for TS recognition, consists of four convolutional layers, each followed by ReLU activation functions to establish non-linearity [25]. After every two convolutional layers, a MP layer with a pool size of 2×2 reduces the spatial dimensions while retaining important features. BN is applied after each MP layer to stabilize training and improve convergence by normalizing the activations. The extracted features are then flattened and passed through a dense layer with 512 units, subsequently BN is applied followed by a dropout layer to avoid overfitting [26]. Finally, the output layer consists of 43 neurons with a SoftMax activation function, corresponding to the 43 TS classes. This architecture effectively balances extraction and dimensionality reduction, ensuring robust performance in TS recognition tasks.

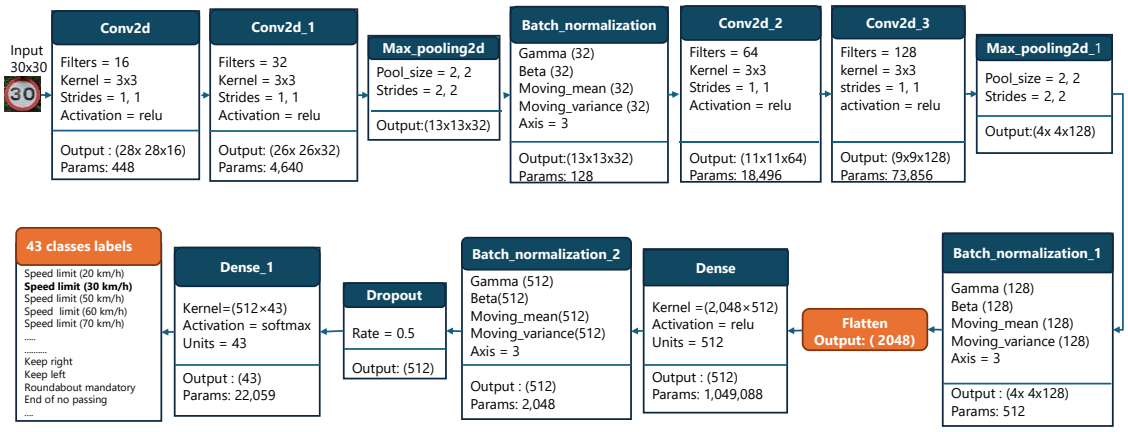


Figure 3. Proposed CNN sequential architecture

2.2. Traffic sign datasets

The dataset consists of 805 images, containing a total of 1,216 TS. Of these, 732 images originate from German TS datasets GTSDB, while 73 images were collected from Moroccan roads environments [2]. TS were categorized into five groups based on color and shape, as detailed in Table 1. The distribution of these categories is illustrated in Figure 4. Roboflow tool [27] was used to label TS datasets, subsequently, data augmentation techniques were employed on the labeled datasets, such as: maximum zoom to 30%, blur up to 0.5px, brightness between -35% and +35%, resulting in a final dataset of 2,403 images, split it to 85% for training, 10% for test, and 5% for validation.

Table 1. TS categories studied

Classes	Examples	Description
Mandatory		Blue rectangular or circular sign with a white pictogram indicating the required action. May have a thin white border.
Yield		Inverted triangle, white background with red border.
Prohibitory		Circular shape, with a red border, white background, and a black pictogram or text indicating the prohibited action.
Warning		Triangular shape pointing upward, with a red border, white background, and a black pictogram indicating a potential hazard or caution ahead.
Stop		Octagonal shape, with a red background, white border, and white text indicating the word "STOP" in French or Arabic language.

For the recognition stage, the GTSRB dataset is used [28], comprising 43 TS classes, split into 39,209 training images and 12,630 test images. Figure 5 shows the numbers of samples across 43 TS classes. Based on datasets distribution in Figures 4 and 5, certain signs, such as prohibitory and speed-limit signs, appear far more frequently, whereas others are rarely represented. This type of imbalance is common in TS datasets, as some signs simply occur more often on real roads. Organizing the signs by their form and color in Table 1, as previously described, helps clarify the overall structure of the dataset despite these unequal class frequencies.

The detection and recognition models were trained using the parameters listed in Tables 2 and 3 respectively. The training was carried out using Google colab GPU (NVIDIA SMI, Cuda version 12 torch 2.0.1 + cu118 CUDA:0, and Tesla T4).

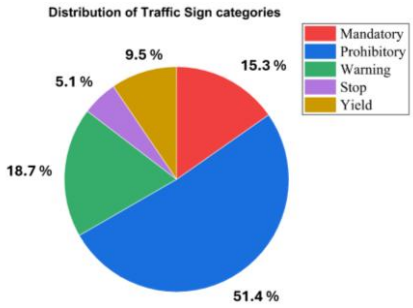


Figure 4. Distribution of TS detection categories

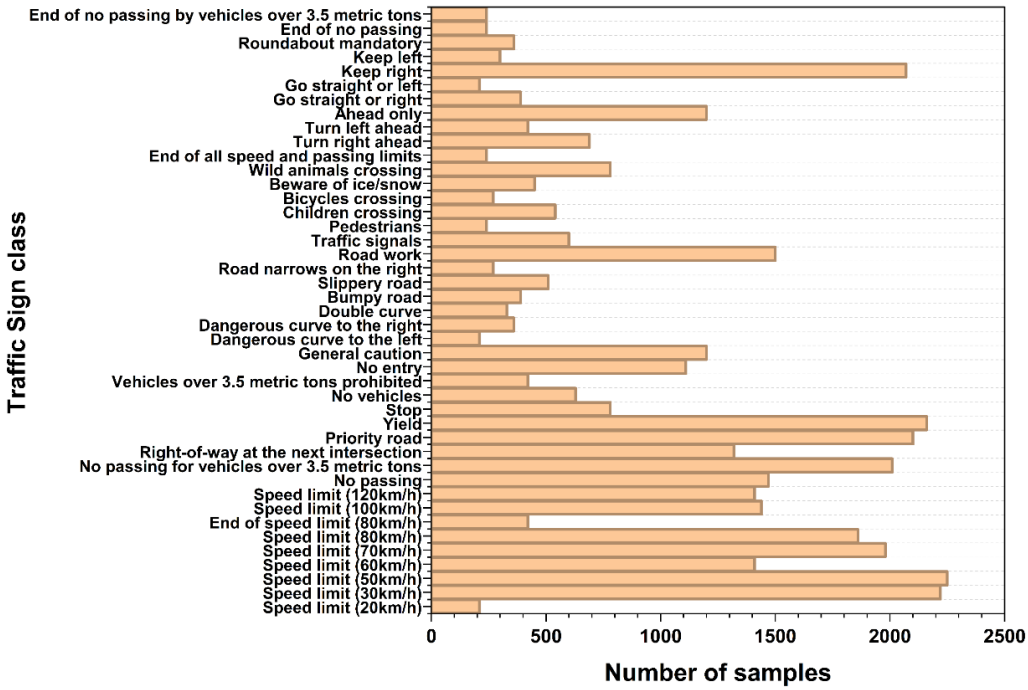


Figure 5. TS recognition classes distribution

Table 2. Training parameters for YOLOv7

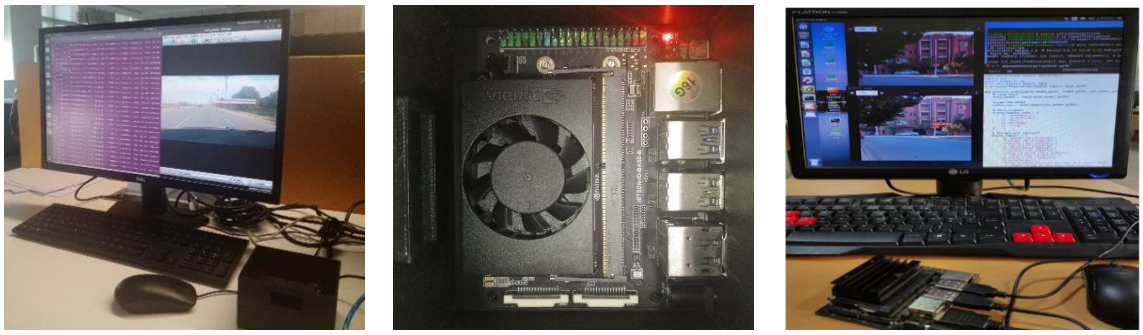
Parameter	Value
Epochs	160
Weight_decay	0.0005
Momentum	0.937
Learning rate	0.01
Batch size	16
Image size	640×640×3
Workers	8

Table 3. Training parameters for CNN

Parameter	Value
Epochs	30
Batch_size	32
Optimizers	Adam
Image size	30×30×3
Learning rate	0.001

2.3. Embedded platforms overview

To offer a practical solution for TS detection and recognition in autonomous vehicles, the developed system is implemented on two embedded system platforms, as illustrated in Figure 6. This figure provides an experimental setup, showcasing the key components of our testbed. Figure 6(a) presents the experimental testbed using the Jetson Xavier NX. It contains the Jetson Xavier NX, Figure 6(b) shows the device in close-up, it is a high-performance AI computer designed for autonomous machines, connected to a display screen. This screen is used to visualize the output of the TS detection and recognition system. Figure 6(c) shows the experimental testbed utilizing the Jetson Nano, a more compact and energy-efficient AI computing platform from NVIDIA, connected to a display screen to monitor the system's output. Both the Jetson Xavier NX and Jetson Nano are heterogeneous computing platforms, combining CPU, GPU, and deep learning accelerators. This architecture is particularly suited for AI applications like the TS detection system. The Xavier NX, with its enhanced computational capabilities, is designed for more demanding AI tasks, while the Nano offers a balance between performance and power efficiency, making it ideal for entry-level AI applications. Table 4 outlines the detailed specifications of both NVIDIA platforms used in the experiments.



(a) (b) (c)

Figure 6. Experimental testbed configurations; (a) testbed with Jetson Xavier NX, (b) close-up of the NVIDIA Jetson Xavier NX module, and (c) testbed with Jetson Nano

Table 4. Embedded platforms specifications

Specifications	Jetson Nano	Jetson Xavier NX
GPU	NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores	384-core NVIDIA Volta GPU with 48 Tensor Cores
CPU	Quad-core ARM Cortex-A57 MP Core processor	6-core NVIDIA Carmel ARM v8.2 64-bit
Memory	4 GB 64-bit LPDDR4, 1600 MHz, and 25.6 GB/s	8 GB 128-bit LPDDR4x, 1600 MHz, and 51.2 GB/s
Power consumption	5-10 watts	10-20 watts
Computing performances	472 GFLOPS	21 TOPS

2.4. Implementation process

To assess the real-time capability of the proposed TS detection and recognition system, a recorded real-world driving video is processed instead of a live camera feed. The algorithm analyzed frames sequentially during playback, each frame is preprocessed using several preprocessing steps like resizing to match the neural network's input requirements and normalization to standardize pixel values. The preprocessed image is then passed through a pretrained deep learning model, YOLOv7 for detection and sequential CNN model for recognition running on the embedded system NVIDIA GPU, which performs TS detection and recognition by analyzing the image features and identifying road signs. The model outputs bounding box coordinates, class labels, and confidence scores for each detected TS. This information is then processed by the system's post-processing algorithm, which filters detections based on confidence thresholds and applies non-maximum suppression to eliminate redundant detections [29]. The processed results are used to generate visual feedback by drawing bounding boxes and labels on the display screen immediately, simulating real-time conditions. This approach ensures reproducibility and controlled testing while demonstrating the system’s potential for real-time deployment.

3. RESULTS AND DISCUSSION

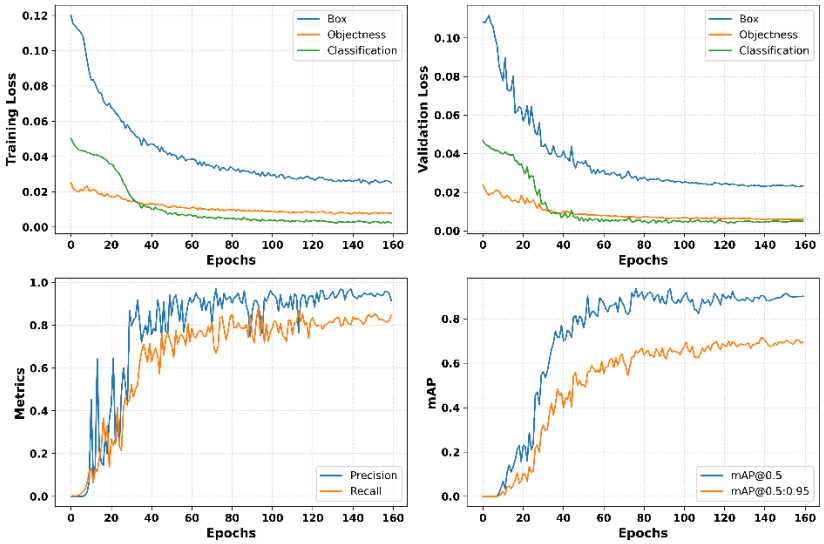
3.1. Model training and evaluation

Both YOLOv7 and YOLOv7 tiny were trained and evaluated on the dataset, Table 5 summarizes their performance. Since YOLOv7 tiny achieved the highest mAP and offers lower model size, making it more suitable for real-time deployment, a detailed analysis of its training and validation behavior is presented in Figure 7. Specifically Figure 7(a), illustrates the progression of both training and validation losses as well as precision, recall, and mAP metrics across epochs. The curves demonstrate steady convergence and indicate that the model generalizes well without overfitting. Figure 7(b) presents the precision–recall curves for all classes on the test set. The tiny YOLOv7 model achieves an overall mAP@0.5 of 90.3%, precision of 96.9% while recall of 80.2%.

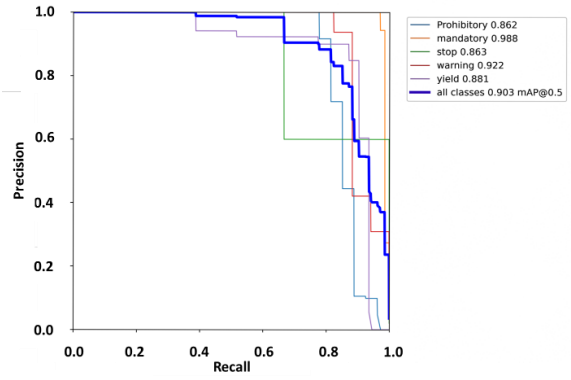
This lower recall is mainly caused by small or distant TS, which produce low confidence detections and often fail to reach the intersection over union (IoU) of 0.5 threshold used. These missed detections reduce recall, even though the model's predictions are typically correct when it does detect a TS, these results indicate that the detector maintains high precision at elevated recall levels across most categories, validating its robustness for TS detection.

Table 5. Evaluation of YOLOv7-Tiny and YOLOv7 algorithm

Method	Precision (%)	Recall (%)	mAP@0.5 (%)	Model size MB)
YOLOv7+kmeans++	88.1	79.1	85.70	74.8
Tiny YOLOv7+kmeans and augmented data	96.9	80.2	90.3	12.3



(a)



(b)

Figure 7. YOLOv7-tiny performance visualization; (a) training and validation dynamics showing loss curves, precision, recall, and mAP across epochs and (b) precision–recall curves for all TS categories on the test set with overall mAP@0.5=0.903

3.2. Deployment results

Experimental evaluation of YOLOv7 models across experimental embedded platforms reveals significant performance variations that highlight the trade-offs between computational power, energy efficiency, and inference speed. Table 6 presents the FPS achieved during the deployment of detection models on the Jetson Nano platform. The lightweight YOLOv7-tiny delivered the highest FPS, demonstrating the substantial impact of model complexity on resource-constrained devices. However, to accelerate inference on NVIDIA Jetson platforms, the trained YOLOv7 models were exported from PyTorch to ONNX format and optimized using TensorRT. The optimization process is done by applying FP16 precision mode to reduce memory usage and improve speed, and building an inference engine tailored to the Jetson hardware. Using this approach, performance improvement is achieved, reaching 18.8 FPS on the same hardware platform, significantly improved throughput while maintaining detection precision.

Figures 8 and 9 show respectively the results of FPS rate and FPS/watt in each power mode of Jetson Xavier NX board. From these results, a clear correlation is observed between available computational resources and inference performance when running the different YOLOv7-tiny and YOLOv7 models. However, for TS detection and recognition applications, power efficiency is indeed a critical consideration, particularly for vehicle-mounted systems where power budget constraints are significant. Findings in Figure 8 indicate that while the 20 W configurations provide the highest absolute FPS performance, they may not be necessary for effective TS detection in many scenarios. The 15 W 6-core configuration emerges as particularly compelling, offering nearly comparable detection performance to higher-power modes while reducing energy consumption by 25%. This power optimization is especially important for electric vehicles where every watt impacts range, and for embedded roadside units operating on limited power supplies. The 10 W 4-core configuration, despite offering the highest FPS/watt ratio as shown in Figure 9, falls below the minimum performance threshold required for reliable real-time TS detection at higher vehicle speeds. This suggests that power efficiency must be balanced against minimum performance requirements rather than maximized at all costs. The results demonstrate that optimal deployment configurations should prioritize sufficient performance within the most efficient power, rather than maximum performance or minimum power consumption in isolation. This approach ensures reliable TS detection while minimizing the impact on vehicle energy systems or infrastructure power requirements, making the technology more viable for widespread adoption in intelligent transportation systems.

Table 6. YOLOv7 deployment results on NVIDIA Jetson Nano

Model	Model format	FPS	FPS end-to-end
YOLOv7 tiny	Pytorch	9.9	9.25
YOLOv7	Pytorch	1.94	1.91
YOLOv7 tiny	TensorRT	**18.8**	**16.61**

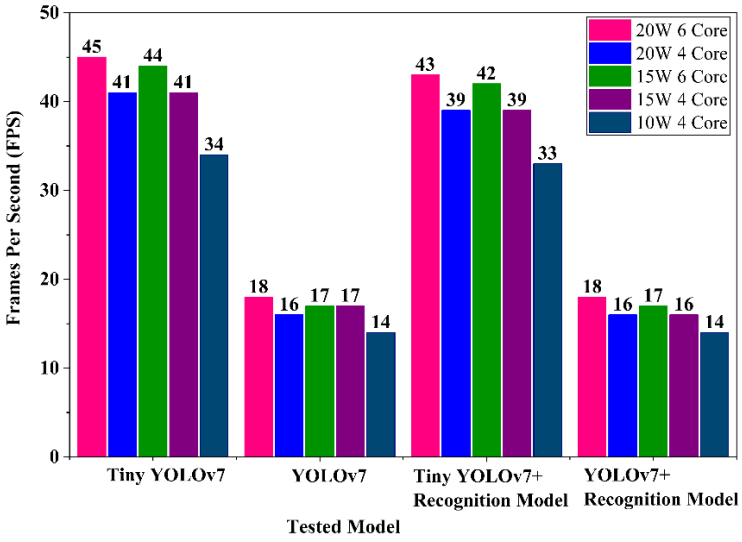


Figure 8. FPS performances of TS detection and recognition on NVIDIA Jetson Xavier NX across different power modes

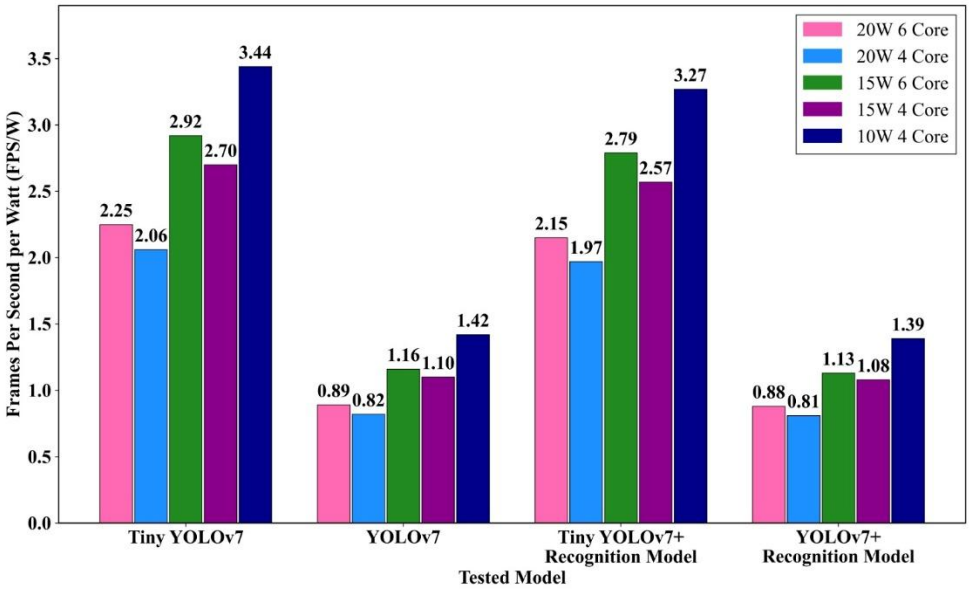


Figure 9. FPS per watt performances of TS detection and recognition on NVIDIA Jetson Xavier NX across different power modes

Hence, testing the recognition sequential model in TensorRT format resulted in an average processing time of 1ms per image, demonstrating its fitting for real time TS recognition. As described in Figure 8, when the detection and recognition models are combined, we observe only a reduction of 2 FPS, which doesn't significantly impact the performance of the integrated system, Figure 10 shows the result of the TS detection and recognition system tested in Morocco roads scenes.

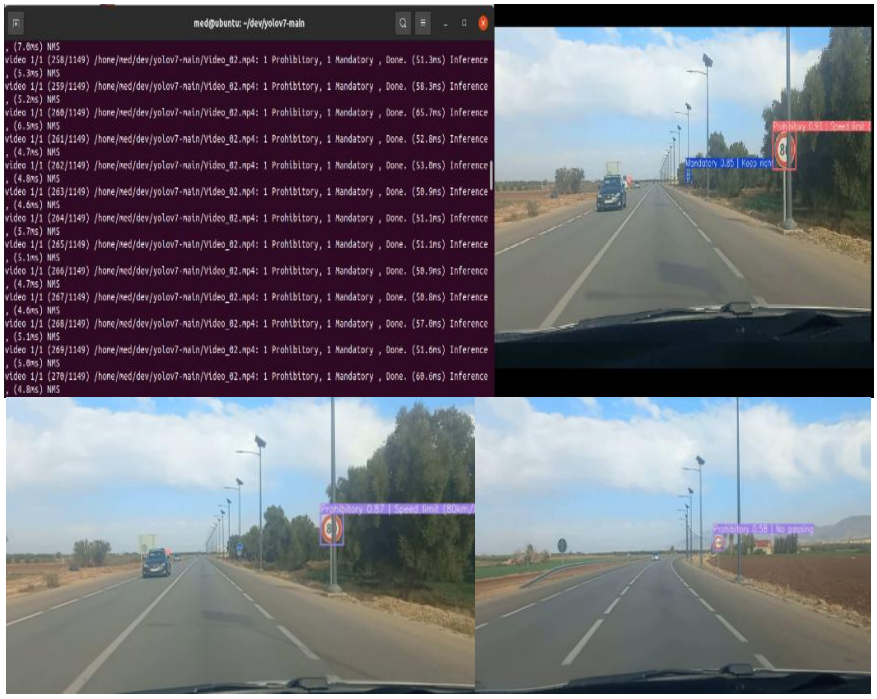


Figure 10. TS detection and recognition applied to Moroccan road: Beni Mellal city test case

3.3. Error analysis and robustness testing

Figure 11 presents the evaluation of robustness under different environmental conditions, including foggy weather in Figure 11(a) and low-light conditions in Figure 11(b).

The detector maintained stable performance across these scenarios, with no significant degradation in precision or recall, indicating strong resilience to illumination and condition changes.

Error analysis was performed on test images collected in Morocco. Representative example is shown in Figure 11(c), where a speed limit sign of 40 km/h, not included in the recognition training classes, was misclassified as the most visually similar existing class, such as the 60 km/h speed limit. This illustrates a limitation in handling out-of-distribution samples. Although Moroccan images were added to improve the detection stage, resulting in strong performance, the classification stage still struggled with certain Moroccan specific classes. Then expanding the training recognition dataset to include further region-specific TS is recommended to improve robustness.

(a) (b) (c)

Figure 11. Qualitative error analysis and robustness testing; (a) correct stop sign detection under foggy conditions; (b) correct prohibitory sign detection under low light conditions; and (c) misclassification of an unseen speed limit 40 km/h sign as speed limit 60 km/h

3.4. Comparison against state-of-the-art works

Table 7 provides a comparative analysis of the inference performance achieved by our deployed YOLOv7 and YOLOv7-tiny models against other models approaches deployed in same or different embedded platforms. The experiments revealed significant performance advantages, with our proposed algorithms consistently outperforming competing methods across both the NVIDIA Jetson Nano and Jetson Xavier platforms. This superior performance was maintained even when using higher input resolutions, further demonstrating the robustness of our implementation.

Table 7. Comparison against state-of-the-art works

Work	Application	Input frame resolution	Method	Embedded platform	FPS
[10]	Object detection	512×512	SSD Mobilenet V2	NVIDIA Jetson Nano	5.6
	Object detection	512×512	SSD Mobilenet V2	Raspberry Pi 4B	4.5
	Object detection	512×512	SSD Mobilenet V2	Raspberry Pi 3B	0.9
[11]	COCO	608×608	PPYOLO-tiny	NVIDIA Jetson Nano	8.1
	COCO	608×608	PPYOLO	NVIDIA Jetson Xavier NX	3.7
[16]	Unmanned surface vehicles	640×640	YOLOv8	NVIDIA Jetson TX2	17.99
(ours)	TS	640×640	YOLOV7 tiny	NVIDIA Jetson Nano	**18.8**
(ours)	TS	640×640	YOLOV7 tiny	NVIDIA Jetson Xavier NX (15W/6 core Mode)	**41.78**

To summarize, the Jetson Xavier demonstrated substantially superior performance over the Nano (with up to 7× improvement in FPS for YOLOv7-tiny), not merely due to raw computational power, but because of its specialized AI acceleration architecture. Xavier's 312-core Volta GPU with dedicated Tensor Cores is specifically optimized for the tensor operations that dominate deep learning inference, allowing it to process the complex backbone in YOLOv7 more efficiently. Additionally, Xavier's unified memory architecture reduces data transfer bottlenecks that particularly affect the Nano during inference. These architectural advantages explain why power-normalized metrics (FPS/watt) show Xavier achieving greater efficiency despite its higher absolute power consumption. Even though, for applications where moderate FPS (~19) is acceptable, then Jetson Nano remains a cost-effective alternative.

4. CONCLUSION

This study investigated real-time TS detection and recognition for ADAS by evaluating YOLOv7 and YOLOv7-tiny models on NVIDIA Jetson Nano and Jetson Xavier NX platforms. The results show a clear correlation between computational resources and detection performance. The optimized YOLOv7-tiny model achieves a strong performance and efficiency trade-off, enabling real-time inference. While the Jetson Xavier NX is able to achieve 43 FPS in the 20 W, 6-core configuration because of its superior hardware capabilities, the model sustained stable performance at 18.8 FPS on the resource-constrained Jetson Nano, and demonstrating that cost-effective platforms can support real-time ADAS when paired with optimized architectures. Future research should focus on recognition dataset expansion by incorporating TS classes specific to Morocco to improve recognition accuracy in local contexts, conducting comprehensive testing under diverse environmental conditions, including adverse weather and varying lighting scenarios.

FUNDING INFORMATION

Authors state there is no funding involved.

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
Imane Taouqi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Mohamed lamane			✓			✓	✓			✓	✓			
Abdessamad Klilou		✓		✓		✓	✓			✓		✓		
Assia Arsalane		✓				✓				✓		✓		
Kebir Chaji					✓	✓				✓		✓	✓	

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

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest.

DATA AVAILABILITY

The data used in this study can be obtained from the corresponding author upon reasonable request.

REFERENCES

[1] M. K. Mani, S. Rajagopal, D. Kavitha, and S. Ramachandran, "Deep Learning-Based Traffic Sign Detection and Recognition for Autonomous Vehicles," in *Digital Twin and Blockchain for Smart Cities*, 1st ed., A. K. Tyagi, Ed., Wiley, 2024, pp. 407–428, doi: 10.1002/9781394303564.ch18.

[2] I. Taouqi, A. Klilou, K. Chaji, and A. Arsalane, "Traffic signs detection and prohibitor signs recognition in Morocco road scene," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 6, pp. 6313-6321, Dec. 2024, doi: 10.11591/ijece.v14i6.pp6313-6321.

[3] J. Zhao, X. Feng, M.-K. Tran, M. Fowler, M. Ouyang, and A. F. Burke, "Battery safety: Fault diagnosis from laboratory to real world," *Journal of Power Sources*, vol. 598, p. 234111, Apr. 2024, doi: 10.1016/j.jpowsour.2024.234111.

[4] Z. Elqabli, O. Kamach, and Y. Chater, "Multimode approach using Reinforcement Learning and Digital Twin for operating mode management," *Production Engineering Archives*, vol. 31, no. 1, pp. 116–128, Mar. 2025, doi: 10.30657/pea.2025.31.11.

[5] S. Liang *et al*., "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345-25360, Dec. 2022, doi: 10.1109/TITS.2022.3158253.

[6] M. Vestias and H. Neto, "Trends of CPU, GPU and FPGA for high-performance computing," in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, Munich, Germany: IEEE, Sep. 2014, pp. 1–6, doi: 10.1109/FPL.2014.6927483.

[7] Z. Yang, K. Adamek, and W. Armour, "Accurate and Convenient Energy Measurements for GPUs: A Detailed Study of NVIDIA GPU's Built-In Power Sensor," in *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, Atlanta, GA, USA: IEEE, Nov. 2024, pp. 1–17, doi: 10.1109/SC41406.2024.00028.
[8] J. Sander, A. Cohen, V. R. Dasari, B. Venable, and B. Jalaian, "On Accelerating Edge AI: Optimizing Resource-Constrained Environments," *arXiv preprint*, 2025, doi: 10.48550/ARXIV.2501.15014.
[9] S. Sonko, E. A. Etukudoh, K. I. Ibekwe, V. I. Ilojianya, and C. D. Daudu, "A comprehensive review of embedded systems in autonomous vehicles: Trends, challenges, and future directions," *World Journal of Advanced Research and Reviews*, vol. 21, no. 1, pp. 2009–2020, Jan. 2024, doi: 10.30574/wjarr.2024.21.1.0258.
[10] A. Zagitov, E. Chebotareva, A. Toschev, and E. Magid, "Comparative analysis of neural network models performance on low-power devices for a real-time object detection task," *Computer Optics*, vol. 48, no. 2, pp. 242–252, Apr. 2024, doi: 10.18287/2412-6179-CO-1343.
[11] J. Zhu, H. Feng, S. Zhong, and T. Yuan, "Performance analysis of real-time object detection on Jetson device," in *2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS)*, Zhuhai, China: IEEE, Jun. 2022, pp. 156–161, doi: 10.1109/ICIS54925.2022.9882480.
[12] X. Long *et al.*, "PP-YOLO: An Effective and Efficient Implementation of Object Detector," *arXiv preprint*, 2020, doi: 10.48550/ARXIV.2007.12099.
[13] E. Civik and U. Yuzgec, "Real-time driver fatigue detection system with deep learning on a low-cost embedded system," *Microprocessors and Microsystems*, vol. 99, p. 104851, Jun. 2023, doi: 10.1016/j.micpro.2023.104851.
[14] 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, p. 1440, Feb. 2024, doi: 10.3390/app14041440.
[15] Y. Luo, Y. Ci, S. Jiang, and X. Wei, "A novel lightweight real-time traffic sign detection method based on an embedded device and YOLOv8," *Journal of Real-Time Image Processing*, vol. 21, no. 2, p. 24, Apr. 2024, doi: 10.1007/s11554-023-01403-7.
[16] A. Haijoub, A. Hatim, A. Guerrero-Gonzalez, M. Arioua, and K. Chougdali, "Enhanced YOLOv8 Ship Detection Empower Unmanned Surface Vehicles for Advanced Maritime Surveillance," *Journal of Imaging*, vol. 10, no. 12, p. 303, Nov. 2024, doi: 10.3390/jimaging10120303.
[17] M. Lopez-Montiel, U. Orozco-Rosas, M. Sánchez-Adame, O. Montiel, K. Picos, and J. J. Tapia, "Traffic Sign Classification Using Real-Time GPU-Embedded Systems," *SN Computer Science*, vol. 7, no. 1, p. 12, Dec. 2025, doi: 10.1007/s42979-025-04634-6.
[18] "jetson nano'. [Online]. Available: https://www.nvidia.com/en-eu/autonomous-machines/embedded-systems/jetson-nano/product-development/. (Accessed: May 12, 2025).
[19] "xavier'. [Online]. Available: https://www.nvidia.com/en-eu/autonomous-machines/embedded-systems/jetson-xavier-nx/. (Accessed: May 12, 2025).
[20] O. N. Neamah, T. A. Almohamad, and R. Bayir, "Enhancing Road Safety: Real-Time Distracted Driver Detection Using Nvidia Jetson Nano and YOLOv8," in *2024 Zooming Innovation in Consumer Technologies Conference (ZINC)*, Novi Sad, Serbia: IEEE, May 2024, pp. 194–198, doi: 10.1109/ZINC61849.2024.10579437.
[21] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-Time Apple Detection System Using Embedded Systems with Hardware Accelerators: An Edge AI Application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020, doi: 10.1109/ACCESS.2020.2964608.
[22] R. Hakani and A. Rawat, "Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano," *Drones*, vol. 8, no. 11, p. 680, Nov. 2024, doi: 10.3390/drones8110680.
[23] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint*, Jul. 2022, doi: 10.48550/arXiv.2207.02696.
[24] R. Zhu, H. Jin, Y. Han, Q. He, and H. Mu, "Aircraft Target Detection in Remote Sensing Images Based on Improved YOLOv7-Tiny Network," *IEEE Access*, vol. 13, pp. 48904–48922, 2025, doi: 10.1109/ACCESS.2025.3551320.
[25] S. Ciurescu *et al.*, "AI in 2D Mammography: Improving Breast Cancer Screening Accuracy," *Medicina*, vol. 61, no. 5, p. 809, Apr. 2025, doi: 10.3390/medicina61050809.
[26] V. Diukarev and Y. Starukhin, "Proposed Methods for Preventing Overfitting in Machine Learning and Deep Learning," *Asian Journal of Research in Computer Science.*, vol. 17, no. 10, pp. 85–94, Oct. 2024, doi: 10.9734/ajrcos/2024/v17i10511.
[27] "roboflow'. [Online]. Available: https://roboflow-com.translate.goog/?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=sc. (Accessed: Dec. 09, 2024).
[28] "gtsrb'. [Online]. Available: https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign. (Accessed: Sep. 20, 2024).
[29] V. Biradar and K. C. Gull, "Multiple Object Detection and Tracking Using Deep Learning Framework with Non-Maximum Suppression," *Multiple Object Detection and Tracking Using Deep Learning Framework with Non-Maximum Suppression*, vol. 18, no. 1, pp. 106–117, Feb. 2025, doi: 10.22266/ijies2025.0229.09.

BIOGRAPHIES OF AUTHORS

Imane Taouqi received the master's degree in communication systems and embedded electronics from the Department of Electrical Engineering at Abdelmalek Essaadi University, Morocco, in 2020. Her area of research includes artificial intelligence, computer vision, image processing and embedded systems. She is currently a Ph.D. student in the fields of embedded systems and artificial intelligence in Faculty of Sciences and Technology, University of Sultan Moulay Slimane, Beni Mellal Morocco. She can be contacted at email: imane.taouqi@usms.ma.

Mohamed Lamane received an Engineer's Degree in 2021 from Sultan Moulay Slimane University, Beni Mellal, Morocco. He is currently pursuing the Ph.D. degree in Mathematics and Applied Physics. He is currently involved in research on improving the classification of targets detected by FMCW radar operating on millimeter waves. He can be contacted at email: com.lamane@gmail.com.

Abdessamad Klilou received an engineer's degree in 2010 and a Ph.D. degree in 2016 from the University of Cady Ayyad, Marrakech, Morocco. Since 2017, he is a professor at the Department of Electrical Engineering in the Faculty of Sciences and Technology, University of Sultan Moulay Slimane, Beni Mellal, Morocco. His area of research includes on parallel and real time optimization of signal processing algorithms on multi-core and multi-processors parallel machine. He can be contacted at email: a.klilou@usms.ma.

Assia Arsalane received an engineer's degree in Electrical Engineering from the National School of Applied Sciences of Khouribga in 2014 and a Ph.D. degree in 2019 from the University of Hassan I, Settat, Morocco. Since 2020, she is an assistant professor in the Department of Mechatronics at the Higher School of Technologies, University of Sultan Moulay Slimane, Beni Mellal Morocco. Her area of research includes artificial intelligence, machine vision, image processing, data analysis, and embedded systems. She can be contacted at email: arsalan.assia@gmail.com.

Kebir Chaji is a professor of electrical engineering at Faculty of Sciences and Technology, University of Sultan Moulay Slimane, Beni Mellal Morocco, since 1996, working in the field of electrical engineering with a focus on numerical modeling and inverse problems applied to thermal systems. He can be contacted at email: kebir.chaji@gmail.com.