UNIVERSITY OF NIS FACULTY OF MECHANICAL ENGINEERING

THE TENTH INTERNATIONAL CONFERENCE
TRANSPORT AND LOGISTICS
https://doi.org/10.46793/TIL2025.235C

til 2025

PERFORMANCE COMPARISON OF EXPLAINABLE ARTIFICIAL INTELLIGENCE IMPLEMENTATION ON TWO EDGE DEVICES FOR A TRANSPORT ENGINEERING PROBLEM

Emilija Ćojbašić [1]***, Maša Despenić*** [2]***, Dejan Rančić*** [1]***,***
Dragan Stojanović [1]***, Ivan Ćirić*** [2]***, Žarko Ćojbašić*** [2]

[1]Faculty of Electronic Engineering, University of Niš, Serbia
[2]Faculty of Mechanical Engineering, University of Niš, Serbia

ORCID iDs:	
Emilija Ćojbašić	https://orcid.org/0009-0008-3843-6703
Maša Despenić	https://orcid.org/0000-0002-5399-2870
Dejan Rančić	https://orcid.org/0000-0002-9445-7700
Dragan Stojanović	https://orcid.org/0000-0003-4893-3814
Ivan Ćirić	https://orcid.org/0000-0003-0430-8937
Žarko Ćojbašić	https://orcid.org/0000-0002-4581-1048

Abstract

The integration of Explainable Artificial Intelligence (XAI) into edge computing systems is increasingly important for safety-critical transport engineering. This paper presents a performance comparison of the YOLO deep learning algorithm for emergency vehicle recognition combined with the Grad-CAM XAI method. Implementations were tested on two edge devices, the Raspberry Pi with AI accelerator module and the NVIDIA Jetson Orin Nano. We analyze performance, power consumption, and cost of the solution, with additional focus on accuracy and deployability. Grad-CAM is assessed for its ability to generate meaningful visual explanations under resource-constrained conditions. Experimental results highlight differences in usability and explanation quality between devices. The findings provide guidance on selecting suitable platforms for real-time, explainable transport engineering applications and contribute to balancing transparency and efficiency in XAI-enabled edge AI solutions for intelligent transportation systems.

Keywords: *Explainable Artificial Intelligence (XAI); YOLO; Grad-CAM; Edge AI; Transport Engineering.*

1 INTRODUCTION

The increasing integration of AI into Intelligent Transportation Systems (ITS) has elevated the importance of not only highly efficient perception models, but also transparent and interpretive decision-making. Safety-critical transport applications that include road user recognition, hazard detection, and intersection control require explainable, real-time outputs to support accountability, regulatory compliance, and operator trust. Recent studies emphasize that XAI plays a crucial role in improving transparency and understanding of model behavior in transportation safety contexts, underscoring the need for interpretable and responsible AI systems in road-traffic applications [1]. Edge-based processing has therefore become essential in latency-sensitive ITS implementations. Performing inference close to the data source reduces communication delays and limits the exposure of raw sensor data to cloud servers, supporting both real-time responsiveness and privacy preservation [2, 3]. However, deploying XAI mechanisms directly on embedded hardware introduces additional challenges. Modern edge devices must execute both object-detection models and explanation algorithms under strict constraints of processing power, memory, and energy consumption [4].

A key difficulty in this context is the performance–transparency trade-off. Deep-learning detectors such as You-Only-Look-Once (YOLO) require substantial computational throughput to maintain real-time inference, while post-hoc explanation methods such as Gradient-weighted Class Activation Mapping (Grad-CAM) add further overhead to the processing pipeline. When both tasks operate concurrently on resource-constrained hardware, maintaining an acceptable balance between inference speed, power efficiency, and explanation quality becomes significantly more demanding [5]. As a result, achieving fast predictions together with meaningful visual explanations on edge platforms remains an open technical challenge for edge-based ITS applications.

In current edge-AI deployments, two dominant hardware approaches have emerged: a modular architecture, where devices such as the Raspberry Pi rely on external NPUs or accelerators to achieve practical inference speeds, and an integrated architecture, exemplified by the NVIDIA Jetson family, which incorporates GPU-accelerated deep-learning capabilities directly into the platform [4, 6]. These approaches differ in performance, power efficiency, and overall system cost, making their comparison essential for real-time, explainable analytics in transport engineering. Although this study focuses on perception and explainability within transport engineering, similar tendencies toward integrating machine learning to achieve real-time efficiency can be observed across other engineering domains [7].

The core contribution of this paper is a rigorous performance comparison of a unified XAI-enabled pipeline: YOLO for emergency vehicle detection combined with Grad-CAM for visual explanations, implemented on both platforms. Unlike prior studies that analyze inference performance in isolation, this work evaluates combined metrics of accuracy, power consumption, explanation overhead, and system cost. The findings contribute to understanding the practical feasibility of explainable, real-time detection on edge platforms and define the hardware factors that play a decisive role in transport-engineering applications.

The remainder of this paper is structured as follows: Section 3 describes the experimental setup and methodology, including the hardware configurations of both edge platforms, the optimisation workflow for the YOLO model, and the integration of Grad-CAM into the inference pipeline.

Key comparative findings arising from the evaluation of detection performance, power consumption, explanation quality, and cost are examined in Section 4. Section 5 concludes the paper by outlining the main insights, limitations of the study, and opportunities for future research in XAI-enabled edge systems for transport engineering.

2 BACKGROUND AND RELATED WORK

The integration of machine-learning and XAI techniques into transportation engineering has shown clear potential for improving safety, interpretability, and trust in automated decision-support systems [8]. These methods help reveal which factors most strongly influence model outputs, a capability that is becoming increasingly important for transparent and accountable decision-making in modern intelligent transportation systems [9].

Accordingly, edge computing has emerged as a key enabling technology. By processing data directly at roadside units or embedded hardware, edge-based systems can deliver real-time perception without depending on cloud connectivity. Recent work demonstrates that vehicle-detection and traffic-counting pipelines can be deployed efficiently on low-cost edge devices while maintaining high accuracy and throughput [10]. This design reduces communication latency, mitigates privacy concerns associated with transmitting raw video streams, and supports operation in bandwidth-constrained environments. Practical deployments already exist on platforms such as Raspberry Pi, Jetson Nano, and Jetson Orin for tasks including vehicle counting, pedestrian monitoring, and traffic-violation detection [4, 11].

For object detection in these systems, the YOLO family has become the model of choice due to its single-stage architecture and ability to meet sub-100-ms inference requirements on embedded hardware. Variants such as YOLOv5, YOLOv7-tiny, and YOLOv8-nano are frequently benchmarked for vehicle detection, traffic-sign recognition, and broader road-scene understanding in resource-limited deployments [12-14]. In parallel, Explainable AI techniques have been increasingly adopted to enhance transparency in computer-vision systems. Grad-CAM is widely used as a post-hoc interpretability method for CNN-based detectors, producing spatial heatmaps that highlight the image regions most influential to the model's predictions [15]. Its intuitive visual output makes it a practical choice for interpreting YOLO-based detections in safety-critical transport applications [16].

Performance benchmarking of edge devices has also received growing attention. Existing studies consistently show that integrated GPU-accelerated platforms such as the NVIDIA Jetson series achieve substantially higher throughput and similar energy efficiency compared with modular single-board computers like the Raspberry Pi. YOLO-based pipelines typically reach real-time performance (20–30 FPS) on Jetson Nano or Orin, while the same models often operate below 5–10 FPS on Raspberry Pi, even when external NPUs are added to improve acceleration [4, 17]. These findings indicate significant differences in compute density, hardware-software integration, and power budgeting between edge-AI architectures. Despite these advances, a key research gap remains. Previous studies have tended to evaluate YOLO inference performance and Grad-CAM explainability separately, often under different hardware conditions and experimental assumptions.

What is missing is a unified analysis that examines how detection speed, accuracy, energy usage, and the computational overhead of Grad-CAM jointly behaves when the same perception-and-explainability pipeline is deployed across heterogeneous edge platforms in a transportation-engineering context. Addressing this gap provides the central motivation for the present study.

3 EXPERIMENTAL SETUP AND METHODOLOGY

In our research, two popular and powerful Edge AI devices have been used. The first used device, the NVIDIA Jetson Orin Nano, is a compact edge-AI module with a 6-core Arm Cortex-A78AE CPU, an Ampere GPU (1,024 CUDA and 32 Tensor Cores), 8 GB LPDDR5 memory, and a 7–25 W power envelope. In our experiments it delivers up to 40 TOPS of AI inference performance under Ubuntu 22.04 with JetPack 6.2, which includes CUDA, cuDNN and TensorRT for optimized deep-learning workloads.

The second used Edge AI device, the Raspberry Pi 5 configuration, combines a quad-core 2.4 GHz Arm Cortex-A76 CPU, 2 GB LPDDR4X RAM, active cooling, an official Camera Module 3 with a 12 MP Sony IMX708 sensor and autofocus, and a Hailo AI accelerator board, version Hailo 8L. Running Raspberry Pi OS with the Hailo software stack, this setup provides 13 TOPS of dedicated NPU compute for real-time, camera-based edge-AI inference while maintaining low power consumption suitable for embedded deployment. Experimental setup with both devices is shown in Figure 1.

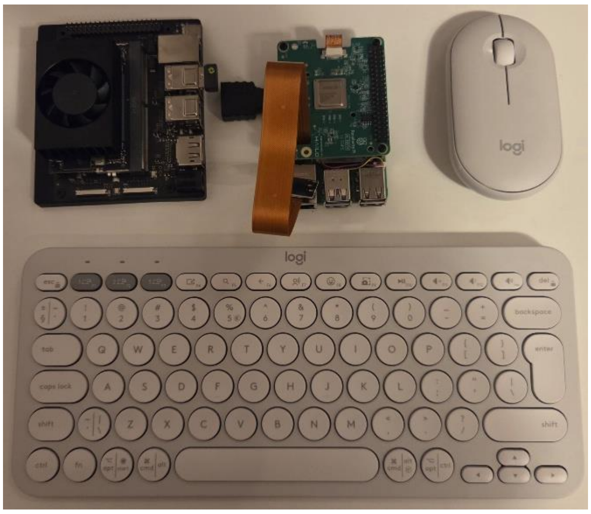

***Fig. 1** Experimental setup with two Edge AI devices*

In transport engineering problem considered, deep learning YOLO algorithm-based detection of emergency vehicles is used, in order to facilitate ITS traffic regulation automatic decision, with XAI integration providing transparency in decisions. The Emergency Vs Non Emergency Vehicles dataset has been used in our experiments [18], which consists of vehicle images categorized into two classes - emergency and non-emergency - for supervised image-classification tasks. It includes 1,370 files with a total size of about 238 MB, providing a reasonably sized corpus for training and evaluating deep-learning models. The images show vehicles in typical traffic conditions, enabling models to learn visual cues that distinguish emergency vehicles (e.g. ambulances, police cars, fire trucks) from ordinary civilian vehicles. The data are organized in a way that supports straightforward use in common computer-vision pipelines, including CNN-based classifiers. Overall, the dataset is well suited as a benchmark for binary classification experiments on emergency-vehicle detection in intelligent transportation and traffic-management applications.

The vehicle detector is based on the YOLOv8n architecture, a compact one-stage convolutional neural network designed for real-time operation on resource-constrained edge hardware. Structurally, it follows a lightweight design with a convolutional backbone, feature pyramid neck, and multi-scale detection head, enabling efficient extraction of hierarchical features and robust detection of vehicles at different distances and sizes in the scene. In this study, the network is trained for emergency vehicle and traffic-scene detection and exported to the ONNX format, enabling an identical inference pipeline on both the NVIDIA Jetson Orin Nano and the Raspberry Pi 5 + Hailo-8L platforms.

Input images were resized to 256×256 pixels size using aspect-ratio-preserving letterboxing: the longer side of the image was scaled to 256 pixels, and the remaining space on the shorter side was padded symmetrically with black pixels to maintain a square input without cropping, which is demonstrated in Fig. 2.

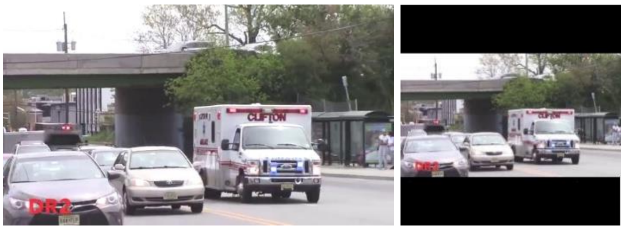

Fig. 2 *Letterbox resize of input images to 256 x 256 size*

In the Jetson Orin Nano configurations, inference utilized both baseline PyTorch and the TensorRT-optimized pipeline. TensorRT significantly accelerated performance via precision calibration, operator fusion, and GPU acceleration. This combined impact of optimized preprocessing, quantized weights, and hardware-specific tuning clearly demonstrates the necessity of aligning data preparation with the underlying inference architecture to maximize system responsiveness.

Custom wrappers were developed to enable seamless use of XAI Grad-CAM technique in combination with the YOLO model, allowing flexible experimentation and evaluation. Implementing these methods on the Jetson platform required targeted adjustments and performance optimizations, including installing a CUDA-enabled PyTorch build to fully utilize GPU acceleration. Grad-CAM, originally intended for CNN-based classification, had to be reworked for object detection by incorporating YOLO's multi-scale feature maps and bounding box logic. The key modification involved capturing high-level activation maps from the model's second to last convolutional layer prior to inference and then applying the standard Grad-CAM computation to those features.

Deploying models on the HAILO-8L imposes specific constraints, requiring models to be exported as ONNX files and processed using the Hailo Model Zoo and TAPPAS SDK. These tools handle INT8 quantization and compilation into a chip-specific binary. Due to strict layer compatibility, model adaptation or retraining is often necessary. With limited debugging tools, this pre-processing is crucial, going beyond simple formatting to ensure the model is robust for real-time inference.

Performance of DL YOLO + XAI model on both devices was critically assessed by several criteria including inference speed, power consumption, detection accuracy, requirements, edge deployment suitability, flexibility, ease of implementation and others.

4 COMPARATIVE RESULTS AND DISCUSSION

Performance of the YOLO model with integrated XAI Grad-CAM layer for both edge devices considered is presented in Figure 3. Performance in Fig. 3 is average of multiple runs with multiple batch sizes, where relative speed is presented taking core Raspberry Pi 5 without acceleration as a reference.

When paired with the Raspberry Pi 5, the Hailo-8L typically demonstrates higher raw object detection throughput (FPS) for quantized INT8 vision models like YOLOv8 compared to NVIDIA Jetson Orin NANO, but when XAI is integrated Jetson's power provides advantage in perfromance. Both devices demonstrated capability to run DL + XAI platform in ITS deployment scenario, where in future experiments and applications a camera would provide live traffic images in real world scenario.

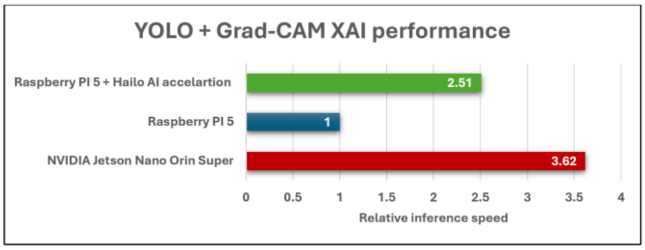

Fig. 3 *Computational performance of the same model with XAI integrated on two Edge AI devices, average from multiple runs and batch sizes*

Model output for single dataset image annotating emergency and non-emergency vehicles and Grad-Cam generated heatmap for an image from test set generated with Jetson device are both presented in Figure 4, (a) demonstrating YOLO model performance and (b) showing that XAI method provide useful visual explanations of recognized emergency vehicles in real world traffic scenarios.

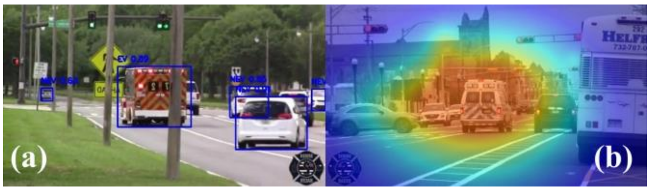

Fig. 4 *(a) YOLO model annotated emergency and non emergency vehicles and (b) Grad-CAM generated heatmap illustrating key decision-making regions*

In Table 1 all findings regarding complex performance comparisons for two edge devices are summarized. While there is no clear winner, it could be concluded that both devices are suitable for the task considered while demonstrating differences that could prove crucial for specific application scenario considered.

Table 1 Performance and suitability comparisons for two edge AI devices in DL + XAI ITS tasks

Category	Comparison and verdict
Performance	NVIDIA Jetson Orin Nano is more powerful than Raspberry Pi 5 with Hailo AI accelerator in DL + XAI ITS tasks, but both are real time capable and results are task and optimization dependent. XAI integration favorizes Jetson to the contrary to YOLO only tasks.
Power consumption	Raspberry Pi 5 with Hailo AI accelerator is more energy efficient than NVIDIA Jetson Orin Nano in DL + XAI ITS tasks, especially when Jetson runs in Super mode, but both are running at relatively low power range 5-25W.
Price	Raspberry Pi 5 with Hailo AI accelerator is somewhat cheaper than NVIDIA Jetson Orin Nano (~20%) but both are in the same price range and are low-cost efficient and applicable solutions for DL + XAI ITS tasks.
Temperature	NVIDIA Jetson Orin Nano is more stable than Raspberry Pi 5 with Hailo AI accelerator in DL + XAI ITS tasks, but both run well as in our experiments both have active cooling while it should be noted that Jetson uses more power.
Deployment suitability	Raspberry Pi 5 with Hailo AI accelerator is somewhat more deployable and easier to setup with alternative power supplies (e.g. solar + battery) in deployment scenarios, but NVIDIA Jetson Orin Nano is more powerful and reliable edge device for DL+ XAI ITS tasks.
Ease of use	Both require significant skills and are not exactly user friendly, but NVIDIA Jetson Orin Nano has more polished environment and integrated CUDA/TensorRT software stack while Raspberry Pi 5 with Hailo AI accelerator has strong community support.
Quality of XAI integration results	Both devices provide good Grad-CAM outputs on traffic scenes, with clear and relevant visual heatmaps regardless of the platform hardware-specific processing.

Compared to the more cost-efficient Raspberry Pi 5 + Hailo 8L configuration, which delivers 13 TOPS of very energy-efficient NPU compute for real-time camera-based inference at low power suitable for embedded deployment, the Jetson Orin Nano offers richer CUDA/TensorRT software stack and higher peak performance up to 40 TOPS in its maximum "super developer mode" power configuration (25 W performance mode), but this comes at the expense of somewhat higher hardware cost and a wider 7–25 W power envelope.

Our empirical conclusion that Jetson Orin Nano, especially in its 25 W "super developer mode," offers higher peak throughput but at higher cost and power than the more cost- and energy-efficient Raspberry Pi 5 + Hailo configuration is consistent with recent benchmarking studies, which similarly show Jetson-class boards leading in raw YOLO/object-detection performance while Raspberry Pi 5 paired with NPUs (Coral TPU or Hailo-8L) achieves a superior balance of TOPS/W and cost per TOPS for real-time edge inference [19-21].

5 CONCLUSION AND FUTURE WORK

It can be concluded that the NVIDIA Jetson Orin Nano in Super Mode is the higher-performance choice with superior integrated CUDA/TensorRT software stack for seamlessly executing the full YOLO and Grad-CAM pipeline, but its developer kit costs somewhat more and consumes more power. Conversely, the Raspberry Pi 5 + Hailo-8L setup is the more cost-effective and power-efficient alternative, which is ideal for pure vision inference tasks, though the split processing for Grad-CAM results in somewhat slower overall end-to-end XAI performace. Nevertheless, both devices are capable of running demanding DL + XAI tasks in complex ITS scenario, are suitable for edge deployment and have similar cost and power consumption, while performance differences are dependent on the specific tasks (e.g., high-throughput motorways vs. low-volume urban intersections) and optimization.

Considering implications for transport engineering and for traffic management authorities, these results imply that selecting between the NVIDIA Jetson Orin Nano and the Raspberry Pi 5 + Hailo-8L should be framed as a multi-criteria decision problem - jointly optimizing required detection and Grad-CAM explanation latency, robustness, energy consumption, and lifecycle cost for the specific deployment scenario, in line with recent Edge AI hardware-selection methodologies that promote systematic trade-off analysis rather than ad hoc device choice [22].

The main limitation of our research is that it is restricted to a single post-hoc XAI technique (Grad-CAM) applied to one specific YOLO-based emergency vehicle detector and two hardware platforms, so the findings may not fully generalize to other explanation methods, model architectures, or ITS tasks, consistent with recent work showing that Grad-CAM can yield relatively coarse, method-dependent explanations and should ideally be complemented by alternative interpretability approaches [23].

Future research could include development and benchmarking of additional XAI techniques specifically tailored for ITS tasks on both devices, extend experiments to multi-camera ITS deployments, systematically assess long-term operational costs and thermal management under real traffic conditions and combine edge AI approaches with distribted AI to resolve very complex ITS scenarios [24].

Explainability is vital for the next generation of safe and widely adopted Intelligent Transportation Systems (ITS) because it serves as the key link between complex AI decision-making and human trust, providing the transparency needed for rapid operator intervention, regulatory compliance, and ongoing performance improvement in safety-critical ITS scenarios [25].

ACKNOWLEDGMENT

This work has been supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [Grant Numbers: 451-03-137/2025-03/200102 and 451-03-137/2025-03/200109].

REFERENCES

1. Abdulrashid, I., Farahani, R.Z., Mammadov, S., Khalafalla, M., Chiang, W.-C., 2024, Explainable artificial intelligence in transport logistics: Risk analysis for road accidents, *Transportation Research Part E: Logistics and Transportation Review*, 186, 103563
2. Gong, T., Zhu, L., Zhu, L., Yu, F.R., Tang, T., 2023, Edge Intelligence in Intelligent Transportation Systems: A Survey, *IEEE Transactions on Intelligent Transportation Systems*, PP(99), pp. 1–26.
3. Andriulo, F.C., Fiore, M., Mongiello, M., Traversa, E., Zizzo, V., 2024, Edge Computing and Cloud Computing for Internet of Things: A Review, *Informatics*, 11(4), 71.
4. Alqahtani, D.K., Cheema, M.A., Toosi, A.N., 2025, *Benchmarking Deep Learning Models for Object Detection on Edge Computing Devices*. In: Gaaloul, W., Sheng, M., Yu, Q., Yangui, S. (eds) Service-Oriented Computing. ICSOC 2024. Lecture Notes in Computer Science, vol 15404. Springer, Singapore.
5. Vermesan, O., Piuri, V., Scotti, F., Genovese, A., Donida Labati, R., Coscia, P., 2024, *Explainability and Interpretability Concepts for Edge AI Systems*, in book: Advancing Edge Artificial Intelligence, River Publishers.
6. Biglari, A., Tang, W., 2023, A Review of Embedded Machine Learning Based on Hardware, Application, and Sensing Scheme, *Sensors*, 23(4), 2131.
7. Milić, P., Marinković, D., Klinge, S., and Ćojbašić, Ž., 2023, Reissner-Mindlin based isogeometric finite element formulation for piezoelectric active laminated shells, *Tehnički vjesnik*, 30(2), pp. 416–425.
8. Subashka Ramesh, S., Faritha Banu, J., Kavitha, V.R., Ramesh, T., 2025, Enhancing Intelligent Transportation Systems in Smart Cities Using VANETs With Deep Reinforcement Transfer Learning and Explainable AI, *Transactions on Emerging Telecommunications Technologies*, 36(8), e70219.
9. Sariyer, G., Mangla, S.K., Sozen, M.E., Li, G., Kazancoglu, Y., 2024, Leveraging explainable artificial intelligence in understanding public transportation usage rates for sustainable development, *Omega*, 127(4), p. 103105.
10. Dinh, D.-L., Nguyen, H.-N., Thai, T., Le, K.-H., 2021, Towards AI-based traffic counting system with edge computing, *Journal of Advanced Transportation*, 2021(2):1-15.
11. Elmanaa, I., Abdelouahed, S.M., Abouch, Y., Aarab, A., 2023, Efficient roundabout supervision: real-time vehicle detection and tracking on Nvidia Jetson Nano, *Applied Sciences*, 13(13), 7416.
12. Luo, Y., Ci, Y., Zhang, H., Wu, L., 2024, A YOLOv8-CE-based real-time traffic sign detection and identification method for autonomous vehicles, *Digital Transportation and Safety*, 3(3), pp. 82–91.
13. Liang, Z., Wang, W., Meng, R., Yang, H., Wang, J., Gao, H., Li, B., & Fan, J. , 2024, Vehicle and Pedestrian Detection Based on Improved YOLOv7-Tiny. *Electronics*, 13(20), 4010.
14. Xu, J., Du, Y., Yi, Y., Du, H., 2025, An improved lightweight algorithm for traffic sign detection, *Scientific Reports*, 15, 33554.
15. Sieradzki, A., et al., 2024, Explainable AI (XAI) Techniques for Convolutional Neural Networks, *Applied Sciences*, 14(17), 7462.
16. Wang, Y. et al., 2024, Lightweight Vehicle Detection Based on Improved YOLOv5s, *Sensors* 24(4), 1182.
17. Santos, R., Silva, M.C., Rabelo, R.A., 2024, Real-time object detection performance analysis using YOLOv7 on edge devices, *IEEE Latin America Transactions*, 22(10), pp. 799–805.
18. Shitole, R., 2024, *EV vs NEV Annotated Vehicle Image Dataset*. [Dataset] Kaggle. Available at: https://www.kaggle.com/datasets/rohitshitole96/ev-vs-nev-annotated-vehicle-image-dataset/data (Accessed: 15th November 2025).
19. Minott, D., Siddiqui, S. and Haddad, R.J., 2025, *Benchmarking edge AI platforms: performance analysis of NVIDIA Jetson and Raspberry Pi 5 with Coral TPU*, in Proceedings of IEEE SoutheastCon 2025, pp 1384–1389.
20. Ortiz-Garcés, I., Villegas-Ch, W. and Luján-Mora, S. , 2025, Implementation of edge AI for early fault detection in IoT networks: evaluation of performance and scalability in complex applications, *Discover Internet of Things*, 5, 108, pp. 1–33.
21. Rey, L. et al, 2025, A Performance Analysis of You Only Look Once Models for Deployment on Constrained Computational Edge Devices in Drone Applications, *Electronics*, 14(3), 638.
22. Canpolat Şahin, M. and Kolukısa Tarhan, A., 2025, Evaluation and Selection of Hardware and AI Models for Edge Applications: A Method and a Case Study on UAVs, *Applied Sciences*, 15(3), 1026.
23. Ennab M., Mcheick H., 2025, Advancing AI Interpretability in Medical Imaging: A Comparative Analysis of Pixel-Level Interpretability and Grad-CAM Models, *Machine Learning and Knowledge Extraction*, 7(1), 12.
24. Ghasemi, A., Keshavarzi, A., Abdelmoniem, A.M., Nejati, O.R. & Derikvand, T., 2025, Edge Intelligence for Intelligent Transport Systems: Approaches, challenges, and future directions, *Expert Systems with Applications*, 280, 127273.
25. A. Kuznietsov et al, 2024, Explainable AI for Safe and Trustworthy Autonomous Driving: A Systematic Review, *IEEE Transactions on Intelligent Transportation Systems*, 25 (12), pp. 19342-19364.

Contact address:
Emilija Ćojbašić
University of Niš
Faculty of Electronic Engineering
18104 Niš
A. Medvedeva 4
E-mail: emilija.cojbasic@elfak.ni.ac.rs