Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 53(1), 2026, Pages 291–309, DOI: 10.52846/ami.v53i1.2386
ISSN: 1223-6934

A YOLO Analysis for Vehicle Recognition - Detection Improvements, Cropping Strategies and Cascading Architecture

Andrei Gabriel Nascu, Daniel-Gheorghe Gagiu, and Dan Selişteanu

Abstract. With the development of high-resolution cameras and enhanced storage devices, image datasets are increasingly captured with higher spatial dimensions. Many modern object detector architectures, such as the YOLO family, typically operate at a 640x640 resolution. When training on high-resolution images, the rescaling process can drastically reduce the region of interest, especially for low- or medium-sized objects far from the camera. To analyze the impact of rescaling, a six-vehicle dataset was constructed from high-resolution images. An automated preprocessing pipeline is built, that crops each car region, enforcing a minimum size of 640x640 by symmetrically expanding the margins. A refactoring process was applied to the new set of images, resulting in a new dataset centred on the region of interest. A YOLOv10 model was trained for each dataset, with the cropped dataset achieving a higher mAP50 of 0.995 compared to the uncropped version's 0.987. A cascading multilevel model employing both YOLO models was further proposed, with the first model analyzing the initial high-resolution image at a coarser level, cropping the vehicle area, and sending it to the second YOLO model for fine-level analysis. This architecture highlights the importance of preserving fine, detailed features that would otherwise be lost during scaling.

2020 *Mathematics Subject Classification.* Primary 68T45; Secondary 68T07.
Key words and phrases. YOLO, computer vision, car classification, artificial intelligence, multimodel cascading architecture.

1. Introduction

Artificial intelligence is one of the main driving forces of progress across domains, especially in the automotive industry, where new models of smart cars, traffic lights, and radar are often introduced to the market. The common trend in this field is to integrate IoT sensors and cameras to collect data for decision-making, employing Machine Learning and Convolutional Neural Networks techniques for both real-time and non-real-time applications. This process of data collection, sharing and interpretation is closely aligned with each innovation and the emergence of new and improved models ([1], [8], [14]). The evolution of cars is a multilayer process that can be traced according to [2] on multiple branches, from which some of the most important are:

- Becoming more friendly with the users, environments and other traffic participants;
- Integrating new technologies like augmented reality and AI;
- Increasing communications with other cars and servers through a robust decentralized network;

Received January 10, 2026. Accepted May 19, 2026.

- New models have been designed to address issues that could become prevalent in the future, such as an aging population, increased city traffic, and shared transportation.

However, this evolution does not unfold equally over time or across all countries. The driving forces of change in the automotive sector can be traced to several factors outlined in [12], such as factories and businesses that develop new technologies, laws and regulations, infrastructure and technical advancements, customers' purchasing power, and trust in new advancements in the field. Like any new major innovation, autonomous cars will face a few trust issues, mainly stemming from accidents and moral/ethical problems that could pose challenges even for humans, let alone an AI algorithm that only interprets data not emotions. An interesting survey conducted in America in [7], which found that the number of accidents involving smart cars has risen sharply from 4 in 2019 to 1353 in 2023. This sudden rise must also be correlated with the increase in the number of smart cars in traffic and with the fact that most of them (71.2%) had no injuries, and only 4.8% implied cases of death. The ethical problems related to managing accidents are addressed in [9], where a few major principles in line with the European Commission's guidance are outlined, including risk minimization, equal treatment for all people, and establishing a threshold for risk acceptance. As stated in [22], a car must be able to recognize different objects that may appear on the road, such as other cars, people, animals, and fallen trees. For this task, complex, fast image-processing algorithms that can identify objects are required. If a crash is inevitable from an ethical point of view, the algorithm must be able to assign an importance score to each object based on the circumstances to minimize damage and avoid the worst-case scenario.

The advancement of AI in the automotive industry must be supported by the development of cities where emerging concepts such as traffic camera monitoring, fast internet networks, smart traffic lights, and IoT devices that monitor pollution generated by cars are becoming normal features of 21st-century cities ([19], [22]). Smart traffic monitoring can be implemented using a system of interconnected cameras that send data to a central point where AI image analysis algorithms extract information from the collected images. Based on these data, a central authority can track specific cars, improve the management of intersections, evaluate pollution generated by different car models, prioritize emergency vehicle access at intersections, and more [10]. One of the main aspects of traffic management that is relevant to all major cities is maintaining traffic fluidity and avoiding congestion. Numerous analyses have focused on key issues such as smart city systems (cameras, sensors, and central traffic planning) and urban morphology (population distribution and employment patterns). The importance of this research is emphasized by studies showing increased traffic delays. Each delay can be quantified in financial terms; therefore, a method to reduce traffic-related time loss can improve people's lives. A study analyzing the evolving impact of delays states that in Texas, delays increased by approximately 2.8 times between 1982 and 2014 [24]. Another study in the Greater Toronto Hamilton Area indicated that transportation demand doubled between 1986 and 2016. In this context, transportation comprises of two main components: vehicles and passengers. A correlation has been explained between age, gender, social status, and population density, and the relevance of each transportation pattern [13].

In the context of the evolution of AI technology in the automotive sector, this paper proposes a novel multilevel decision architecture based on the You Only Look Once (YOLO) model, a high-speed real-time computer vision framework used for object detection and image segmentation, to detect different types of cars. Two scenarios are compared: the first involves training a YOLOv10 model to detect the region of interest and classify the car type, while the second trains a YOLOv10 model on a cropped version of the initial image that focuses primarily on the car type. A set of performance metrics was established, and the results were further analyzed and compared to assess potential improvements. Additionally, a multi-model approach comprising two cascading YOLO models was proposed to enhance the detection of the original/uncropped dataset. This study evaluated the performance of the proposed architectures for each case using a new database of images featuring six types of cars.

The main contributions of this paper are briefly presented below:

- Creating a new high-resolution database of images with several vehicle models and labelling them using polygons;
- Train two different YOLOv10 models: one on the initial database and the second on a cropped version of the dataset.
- Propose a new multilevel cascading model that integrates both YOLOv10 models.

The rest of the article is structured as follows: After the succinct presentation of AI in the context of self-driving cars and traffic management given in this section, in Section 2 a critical analysis of papers related to car classification, small regions of interest, and composite/ multimodal approaches are provided. Section 3 describes the process of data acquisition and labeling. Section 4 outlines the methodology of the proposed experiments and analyzes the performance metrics, and Section 5 presents the results. In Section 6, discussions are provided and the limitations of the study and potential future work that could complement the findings are analyzed. Finally, Section 7 concludes the paper with the main ideas drawn from this study.

2. Related works

In [3], the authors implemented a multitask learning approach using two main branches in their proposed model. One branch classifies the car model and the other classifies the manufacturer. This architecture significantly improves the classification of cars with similar features (construction design). Both the initial independent classification of the car manufacturer and the model on one side and the newly proposed architecture that combined the classification of the two branches on the other side were tested on an Indonesian dataset, Ina V-Dash, with four carmakers and 10 car types that look quite similar. Multiple options were explored as the backbone of the feature extraction model, including ResNet50, InceptionResNet, VGG16, VGG19, MobileNet, and Inception. Using a Macro-Average of the F1 score across all classes in the dataset, the VGG16 architecture yielded the best results, with both the highest mean score and the largest improvement over the initial two independent models for each task.

In [16], the authors proposed a composite architecture that combines Vision Transformers, YOLO, DINO, and cropping images to identify 13 vehicle classes defined by the Federal Highway Administration (FHWA). One of the main characteristics used by the FHWA to isolate these 13 vehicle types is the number of axles, which can range

from two to more than seven. Based on this information, the initial backbone of the model, namely the Vision Transformers, was augmented with features extracted using a region-of-interest (ROI) detection algorithm focused on wheel positions. Multiple variants of Vision Transformers were tested to determine the one that provided the highest accuracy. The authors evaluated vanilla ViT, Cross-Attention Multi-Scale Vision Transformer, Transformer in Transformer, and Swin Transformer, with the Swin Transformer achieving the highest accuracy of 87.3 on the original images and 92.3 on cropped images with background noise removed. To leverage wheel information for better classification, several ROI algorithms were tested, including Faster R-CNN with ResNet-50, Faster R-CNN with MobileNetV3, YOLOv4, and YOLOR. YOLOR delivered the best results in terms of the mean average precision and frames per second. A self-supervised algorithm, DINO (distillation with no label), was integrated into the architecture to utilize unlabeled data; it was tested with three options: pretrained, fine-tuned, and retrained, with the pretrained version achieving the highest accuracy when combined with a Vision Transformer and YOLOR - the final step involved randomly masking one wheel of the vehicle, akin to adding noise. The final composite model achieved an accuracy of 96.7%, showing a significant improvement over the initial results obtained with only ViT.

An aerial vehicle detection method is described in [18] that uses images captured by drones, planes, or satellites. For the experiment, two image datasets were used: VEDAI with nine classes and VAID with seven classes. A composite architecture comprising five steps was proposed. First, a preprocessing phase is performed on the images, including defogging and gamma correction to enhance image quality. Next, Fuzzy C-means segmentation was applied to distinguish the foreground from the background pixels. In the third step, the YOLOv8 algorithm is used to detect vehicles in each image. The detected bounding boxes were then analyzed using three feature extraction algorithms: SIFT, KAZE, and ORB. Finally, a Deep Belief Network is used to classify each detected vehicle.

In a comprehensive review [17], the authors analyzed two main deep learning architectures for image classification: CNNs and ViTs. To identify the most significant papers in the field, they searched four major databases: Google Scholar, ScienceDirect, Scopus, and ACM Digital Library. After the initial selection, 31.441 articles were identified. To reduce this number and select only the most relevant articles, a set of inclusion criteria was established, considering the recency of the articles (with ViTs being a novel concept introduced in 2020), language of publication, number of citations, and requirement to analyze both architectures within the same paper. Applying these criteria yielded 17 studies, with training datasets spanning healthcare, agriculture, and transport systems. One of the main conclusions of this study is that ViTs, or composite models, generally achieve better accuracy than traditional CNN models, while consuming fewer resources.

A study on the efficiency of recognizing eight car types using a slightly modified YOLO architecture was conducted in [25]. The database was built using fixed cameras on expressways, highways, and tunnels in China, with eight classes: bus, minibus, family sedan, taxi, heavy truck, truck, SUV, and special vehicle, totaling 2844 images. One of the main issues was the uneven distribution of images across classes, with heavy truck having around 800 instances and classes such as bus, taxi, and special vehicle having approximately 200 images each. Another problem was the small

Region of Interest caused by the camera's position, which can significantly degrade the performance of YOLO algorithm. The authors initially used a simple YOLOv5 algorithm and then enhanced it with a Flip-Mosaic data augmentation technique to mitigate class imbalance. The new architecture showed notable improvements in the SUV and Family Sedan classes, with performance metrics such as mAP50 and mAP50:95 across the entire dataset, yielding better results than the initial model.

A new architecture, YOLO-Vit, that integrates a Vision Transformer into a YOLO version 7s model was proposed in [26] to detect regions of interest in infrared images from uncrewed aerial vehicles (UAVs). To reduce the complexity of the initial model, the backbone is replaced with a lightweight, MobileVit network to extract image features. To minimize information loss for small targets and to expand the perceptual field of the resulting feature map, the authors employed a C3-PaNet model composed of a CARAFE upsampling module and C3 structure. The database used in this study was a slightly modified HIT-UAV dataset that included three classes: person, vehicle, and bicycle, with more than half of the targets in the images being 32x32 pixels or smaller. The results showed a 52.6% reduction in the number of parameters compared to YOLOv7s, a 0.9% increase in recall and mAP, a 1.3% increase in F1 score, and a 0.2% decrease in precision.

Two lightweight YOLOv7-derived models, LEAF-YOLO and LEAF-YOLO-N, were proposed in [20], aiming to significantly reduce the number of parameters while maintaining or even surpassing the accuracy of state-of-the-art models. This study focuses on classifying small objects in aerial images, particularly those captured by uncrewed aerial vehicles. The datasets used for training and testing are VisDrone-2019 (10 classes) and TinyPerson (2 classes). The images were unevenly distributed among the classes and varied in resolution from 950x540 up to 2000x1500 pixels. The two models present a comprehensive approach to improvements across the entire architecture, involving modifications to the backbone with the addition of two new blocks, LEAF and MaxPooling Ghost Convolutions, enhancements to the neck using Coord-Blocks, Partial convolutions, LEAF-T, and Spacial Pyramid Pooling Receptive Field Module, and modifications to the head, including a new head specifically for small objects. In addition, the loss function was changed from CioU to InnerShape-IoU. All these modifications resulted in a LEAF-YOLO model with 4.28 million parameters and a LEAF-YOLO-N model with 1.2 million parameters. For evaluation, traditional metrics were computed such as $AP_{.50:.95}$, $AP_{.50}$, $AP_{.75}$ with results of 28.2%, 48.3%, and 27.6% for LEAF-YOLO and 21.9%, 39.7%, and 20.9% for LEAF-YOLO-N.

3. Database construction

3.1. Image acquisition and anonymization. The database comprises 6 vehicle classes: "Opel Astra", "Dacia Logan", "Dacia Spring", "Tesla", "Dacia Sandero", and "Matiz", each with a slightly different number of images, as shown in Table 1. The resulting database contained 1885 images.

The vehicles were produced by four factories: Opel with 326 images of the Astra model, Tesla with the fewest images at 270, Daewoo with 323 images of the Matiz model, and Dacia with a total of 979 images covering the three models analyzed - Logan with 320 instances, Spring with 334 instances, and Sandero with 325 instances. The significance of this aspect lies in the increased similarity of car bodies within the

TABLE 1. Image distribution per car model.

Vehicle name	Opel Astra	Dacia Logan	Dacia Spring	Tesla	Dacia Sandero	Matiz
Number of images	324	316	332	270	322	321

same manufacturer and in the presence of the manufacturer's logo for each vehicle. The images were captured using the rear camera of a Galaxy S23 Plus smartphone with a resolution of 4000x3000 pixels and 72 dpi. The image acquisition process was conducted by the authors of the article between 2024 and 2025, photographing vehicles in Romania, primarily in Craiova, thereby creating a new database. After the data acquisition process was finalized, sensitive information from the images was removed. This step includes blurring operations for car license plates and human faces if they appear in the images. The tool used to mask certain parts of the photographs was Adobe Photoshop version 27.3.1. Examples for each car after this step can be viewed in Fig. 1.

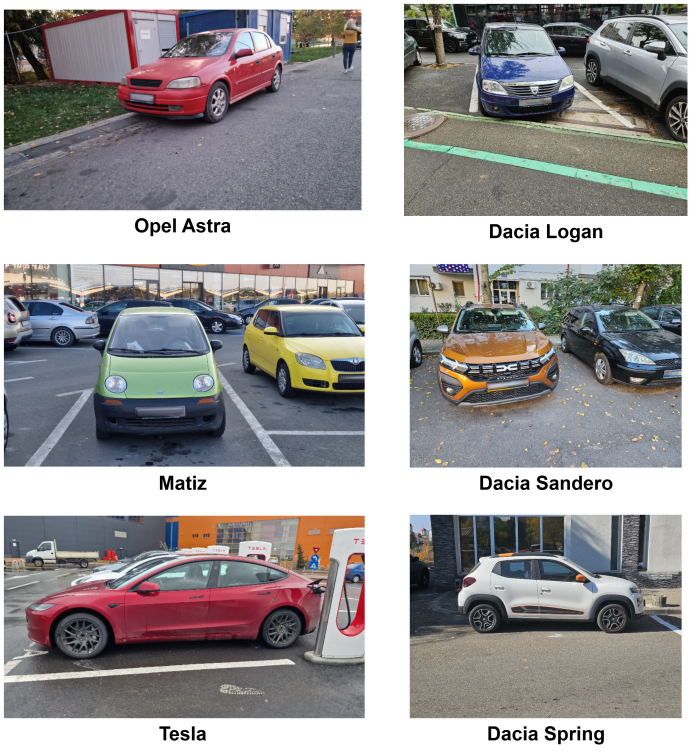

FIGURE 1. Images of each car model were analyzed with blurring of sensitive data.

For a more in-depth exploration of the dataset, three angles were established for the images, taking into consideration the perspective of the viewer (the person who makes the photo/ camera):

- Front, which corresponds to an angle of 0 degrees, represents a view of the car in which the camera sees mostly the front of the car, with little to no lateral perspective;
- 45 degrees, which represents images in which both the front and the lateral part of the car are visible in different proportions;
- This lateral perspective, with an angle close to 90 degrees, offers a good visualization of the car's lateral side, but little to no information about the front of the vehicle.

Each of the above cases can involve a degree of subjectivity and uncertainty, particularly for images that are on the boundary between the two classes. Thus, a classification method must be established.

The methodology for this classification followed a voting scheme: each article author independently assigned a class (front, 45 degrees, lateral) to each image. The class with the highest number of votes is assigned to that image. Following this process, the distribution of images per car and viewing point angle are presented in Fig. 2. This analysis is essential because if any viewing point is missing or underrepresented in the dataset, the AI model, specifically YOLO, used in this study, will not be able to recognize the car from that angle.

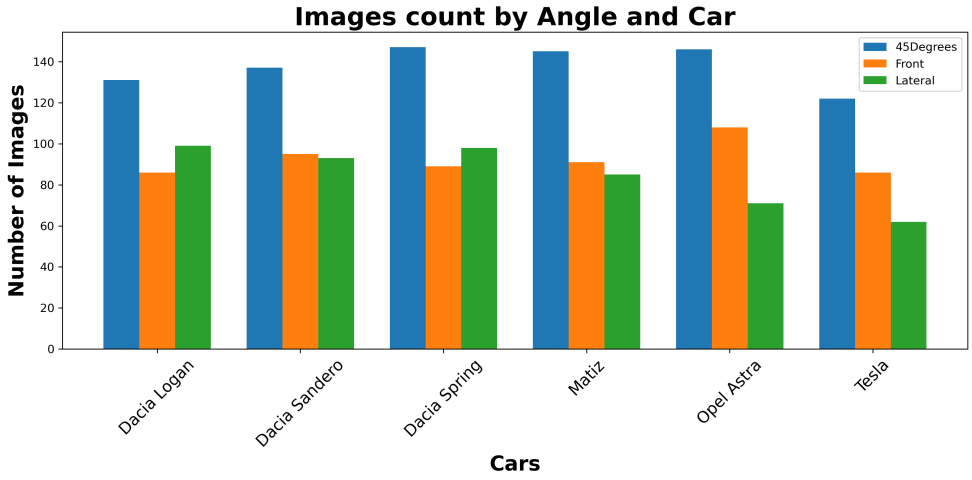

FIGURE 2. Histogram with the number of images divided by vehicle model and point of view of the camera.

3.2. Region of Interest Labeling (ROI). As a labelling option, a polygon was selected to enclose the area of interest in each image, thereby outlining the section of the photograph in which the car was positioned. A YOLO format was used for each annotation, assigning a number to each class (0 = 'Astra', 1 ='Logan', 2 = 'Matiz', 3 ='Sandero', 4 = 'Spring', 5 ='Tesla'). After the class identifier, a set of n normalized points was used to define boundaries of the segmentation polygon. This method

FIGURE 3. Blue polygon that defines the ROI for the Matiz class. Each point from the label file is marked with a filled circle for easier visualization.

enhances the focus on the contours of the labelled objects. It simultaneously allows the training of a model for either bounding box or segmentation, thus maintaining the two main ROI detection options offered by YOLO architectures [11]. The annotation software used was Label Studio version 1.20.0 (https://labelstud.io/), installed within a dedicated Anaconda environment version 25.5.1, using Python version 3.13.5. An annotated image with circles representing each normalized point is shown in Fig. 3. The analyzed image featured an annotation for Matiz (class 2).

To further investigate the distribution of car labels across the dataset, two additional graphics are presented. In Fig. 4a), a heatmap shows the centers of the labels (polygons enclosing vehicles). Most values are in the bottom-left part, but not far from the center of the heatmap, indicating a slight shift in the regions of interest (labelled cars) towards that side of the images in the dataset. The second graphic, Fig. 4b), displays the distribution of width on the OX axis and height on the OY axis, with smaller regions of interest (ROI) (labelled cars) positioned at the bottom right of the graphic and a larger ROI in the upper right corner. Visually, the graph in Fig. 4b) shows that most values are close to the line $x = y$, particularly for smaller x values. As the x values increase, divergence becomes apparent, with a greater scatter of the data points. Another observation is that the data were mostly symmetrical around the line of equality, indicating a high degree of scatter.

An analysis of the space covered by each label in the total image is presented in Table 2. It can be observed that most labels (over 95%) occupy less than 50% of the image, with no labels covering more than 80% of the image, and only one image has a label area between 70% and 80%. This behavior indicates that the labelled

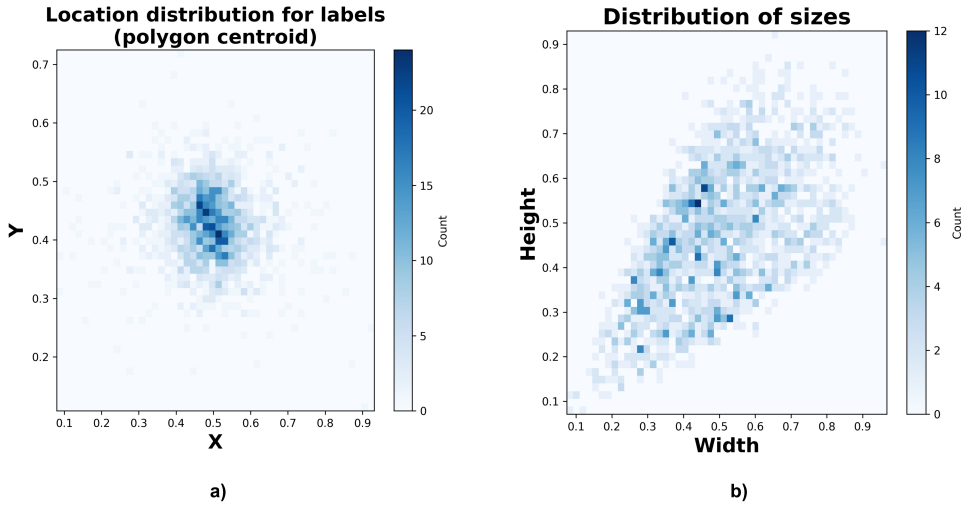

FIGURE 4. Heatmaps with labels distribution for each image. a) The center point of each label b) The width and height of each label.

area typically represents a small portion of the image with large, uninformative backgrounds that lack the relevant features needed to differentiate between car models. This issue is further exacerbated by the YOLOv10 algorithm, which processes images that are resized to 640x640 pixels during training. This is problematic because of the compression and rescaling applied to adapt the original 4000x3000-pixel dimensions to the target 640x640 pixel size. Such rescaling uniformly reduces the entire image, including the key areas of the annotations, leading to a loss of details relevant for identifying cars.

TABLE 2. Label distribution throughout the dataset in relation to the percentage of the image that it covers.

Label percentage from image	0% - 10%	10% - 20%	20% - 30%	30% - 40%	40% - 50%	50% - 60%	60% - 70%	70% - 80%	80% - 90%	90% - 100%
Number of images	260	549	510	359	172	76	16	1	0	0
Percentage from database	13.38 %	28.26 %	26.25 %	18.48 %	8.85 %	3.91 %	0.82 %	0.05 %	0 %	0 %

4. Methodology

Throughout this paper, we utilize the YOLOv10 model [23] to obtain the experimental results. For the first part of the proposed experiment, the dataset was split into 80% training and 20% testing, as shown in Fig. 5A. After the split, the YOLOv10 model

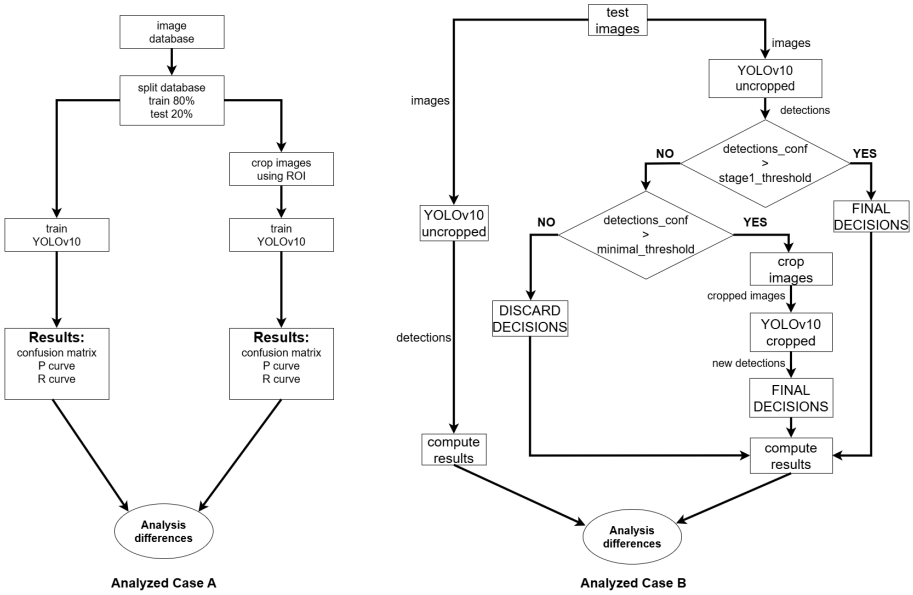

FIGURE 5. A) Left: A straightforward training process is performed on the dataset, Right: A modified, cropped dataset is used for training. B) The two proposed models are compared using the same set of test images. Left: The simple model is tested directly on the dataset, Right: The new architecture combining both YOLOv10 models.

was trained under two scenarios. In both cases, the main model training parameters remained unchanged to ensure consistency in the results and analysis, with the only difference being in the database images. The models were trained using the following set of parameters:

- Image size of 640x640 pixels
- 100 epochs
- 6 classes of cars
- Batch size of 8

In the first scenario, shown in the left branch of Fig. 5A, after splitting the dataset, the YOLOv10 model trains normally and produces a set of performance metrics, such as a confusion matrix, P curve, and R curve. For the second part of the experiment, in the right branch of Fig. 5A, an additional step is included before training begins: cropping the image to the region of interest defined by the YOLO format files. By adding this step before training, we improved the model's ability to analyze more details for each image class by focusing solely on the part of the image that best represents each car type.

For consistency, the same performance metrics were calculated, analyzed, and compared to those from the first scenario. For image cropping in the new step of the second scenario, the desired dimension was 640x640 pixels used by the YOLOv10 model for training. Considering this goal, two main situations can appear:

FIGURE 7. Left: Initial Spring image with two cars labelled. Right: A small, cropped version of the left car from the initial image, which needs to use more background to meet the (640,640) dimension goal.

FIGURE 6. Left: initial image with a Sandero and a polygon highlighting the ROI. Right: cropped version of the same image with a transformed annotation file that identified the new ROI.

- The ROI exceeds the 640x640 pixel objective. In this case, the dimensions of the cropped version were determined by the YOLO file used to annotate the location of the car. This example is shown in Fig. 6, where an initial image of 4000x3000 pixels was transformed into a new version of 2280x1104 pixels.
- The ROI was smaller than the 640x640 YOLO training dimension. In this case, the cropped image will be 640x640 pixels, with the center of the new image set as the ROI center, and the remaining ROI area will be evenly distributed on all sides until the 640 goal is reached. This situation is shown in Fig. 7, where two Spring cars are annotated; one, which is smaller than the desired target, will have its dimensions expanded in all four directions.

Considering w_a and h_a as the width and height of the annotation, respectively the new dimensions for the cropped image w_n and h_n are obtained using the following

approach:

$$(w_n, h_n) = \begin{cases} (w_a, h_a), & \text{if } w_a \geq 640 \text{ and } h_a \geq 640 \\ (640, h_a), & \text{if } w_a < 640 \text{ and } h_a \geq 640 \\ (w_a, 640), & \text{if } w_a \geq 640 \text{ and } h_a < 640 \\ (640, 640), & \text{if } w_a < 640 \text{ and } h_a < 640 \end{cases} \tag{1}$$

For the cropping process to be consistent with the following training step, all annotations must be adjusted to new width and height values for the cropped images. If this modification were not performed, the label files would no longer correctly indicate the ROI for the new dataset. Considering (x_0, y_0) as the normalized values for the top left corner of the cropped rectangle with respect to the initial image, (w_i, h_i) as the width and height of the initial image, and (x_k, y_k) an arbitrary point of the polygon, with $k = \overline{1, m}$ where m represents the number of points of the polygon, then the adjustment for each normalized point of the polygon is performed using the formula below.

$$x' = \frac{x_k - x_0}{w_n / w_i} \qquad y' = \frac{y - y_0}{h_n / h_i} \tag{2}$$

In the first scenario, the initial image database was identical for both branches. However, after the cropping step in the right branch, the models had different training and testing images. To evaluate whether using both models in a joint decision process offers an improvement, a new multimodal cascading architecture is proposed in the right branch of Fig. 5B. Fig. 5B compares the left branch, which solely uses the YOLOv10 model trained on the uncropped dataset, with the new multimodal cascading architecture on the same set of images without initial cropping. The new architecture first employs the same YOLOv10 model on uncropped images to provide a set of detections (bounding boxes and confidence levels). The confidence level is then compared with a stage1_threshold value, and the final decision was entrusted to the initial model if the confidence exceeded this threshold. If the confidence level is lower, a comparison is made with the minimal threshold value.

If the confidence falls below this second threshold, the detection is discarded. If the confidence is higher, the image is cropped using the bounding box from the first YOLOv10 model and sent to the second YOLOv10 model, which is trained on the cropped images, and then delivers the final decision.

The main aspect of this scenario was that the architecture proposed for each branch used the same set of images. Cropping occurs only when cars are identified with a confidence level between the stage1_threshold and the minimal threshold. The YOLOv10 cropped model was then used to better identify the vehicle model by analyzing more distinctive car features. Therefore, the architecture proposed for each branch used the same set of images.

5. Results

A clear and objective basis for the initial evaluation of the impact of cropping on training the YOLOv10 model is provided by the confusion matrices shown in Figure 8 a) for the original high-resolution image dataset and in Fig. 8b) for the modified cropped version of the dataset. In the first case, where all high-resolution images were rescaled during training to the model's input size of 640x640 pixels, the confusion

matrix showed strong diagonal dominance, indicating high accuracy across all six classes analyzed in this study. However, an issue arises with the background class: for a considerable number of images - 20 in this case - the vehicle models are detected in incorrect locations, meaning that true backgrounds are misclassified as one of the analyzed car models, or existing ground truths are not detected. This problem is particularly pronounced in the Tesla and Logan classes, where confusion increases compared to the other classes - 'Astra' with 4, 'Spring' with 3, 'Matiz' with 2, and 'Sandero' with 1. This behavior can be attributed to the loss of fine visual details during image downscaling, especially when the region of interest occupies only a small fraction of the image.

By comparison, the confusion matrix in Fig. 8b), corresponding to the cropped dataset, shows a clear improvement over that in Fig. 8a) for background prediction. The number of background-related errors was drastically reduced across all classes. Most notably, "Tesla" class becomes perfect, with correct predictions increasing from 54 to 59. This improvement reinforces the idea that cropping images around the region of interest preserves sufficient discriminative visual features, thereby strengthening the car prediction model.

A particularly interesting aspect of the two matrices is the false negative cases in which the YOLOv10 model fails to recognize and assign the region of interest to any of the six vehicle classes. This event may occur in one of three ways: no bounding box is detected for that specific region, the confidence of the detected bounding box did not surpass the minimum threshold to be considered viable, or the intersection over union (IoU) value does not reach the minimum required to be recognized as a correct detection. In Fig. 8a), this background-related error occurs 8 times, with the most significant issue occurring in the "Spring" class, where nearly half of these misclassifications occur. Conversely, for the second model, trained on the cropped version of the database, this error is significantly reduced to just one case for the "Logan" class and one for the "Spring" class, as shown in Fig. 8b).

To provide a more in-depth analysis of the model's performance, precision-recall curves were plotted for both YOLOv10 training cases. These graphs serve as complementary evaluation metrics for confusion matrices. The main advantage of this graphical analysis is that it provides a clear view of the trade-off between precision and recall at different confidence levels. In the first case, where the initial unmodified database was used, the aggregate performance, measured by mAP@0.5, was 0.987, as shown in Fig. 9a). This indicates a robust overall performance across all classes, but also suggests that improvements are possible. The Precision-Recall curve for this case indicates a class-dependent degradation at high recall values, where precision declines firstly for "Tesla", and then after recall gets closer to 1 a decrease for "Spring" and "Sandero" can be noticed. Three of the six classes, "Matiz", "Logan" and "Astra"—exhibit nearly perfect scores, with 1, 0.997, and 0.994 for mAP@0.5, respectively, while the underperformers are "Tesla" with 0.968, "Spring" with 0.981 and "Sandero" with 0.984. This suggests that when the YOLOv10 model is tasked with detecting all cars, it becomes less stable and significantly increases the false positives or incorrect localizations of the region of interest.

Compared to the first case, the Precision-Recall curve of the second YOLOv10 model, trained on the cropped dataset, shows a significant improvement, as illustrated in Fig. 9b). The second graph shows a consistent performance across all car classes,

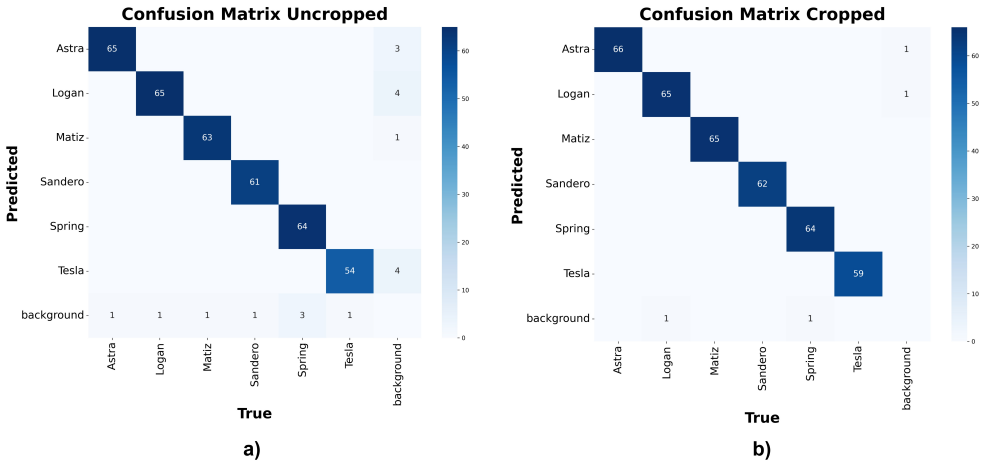

FIGURE 8. a) Confusion matrix for 6 vehicle classes + background for the YOLOv10 on the initial database, b) Confusion matrix for 6 vehicle classes + background for YOLOv10 on the cropped version of the database.

with an mAP@0.5 score of almost 1. This consistency is particularly important for the three vehicle types with the lowest scores, where "Tesla" increases by 3.2%, "Spring" by 1.8% and "Sandero" by 1.6%. In this case, the precision remained close to the maximum value of 1 even when the model was tuned to maximize recall. This indicates minimal performance degradation across different recall levels.

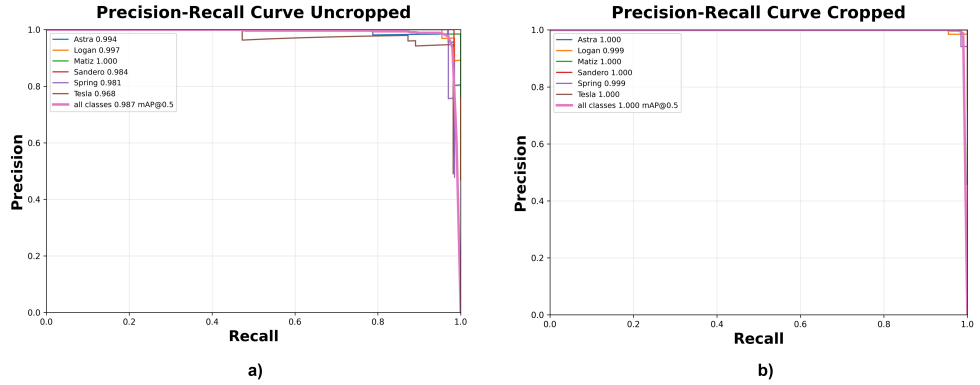

FIGURE 9. a) Precision-Recall Curve for the YOLOv10 model trained on the initial unmodified/uncropped dataset b) Precision-Recall Curve for the YOLOv10 model trained on the cropped image dataset.

By running the proposed new multimodal cascade architecture on the same set of images used for a simple YOLOv10 model, the experiment described in Fig. 5B)

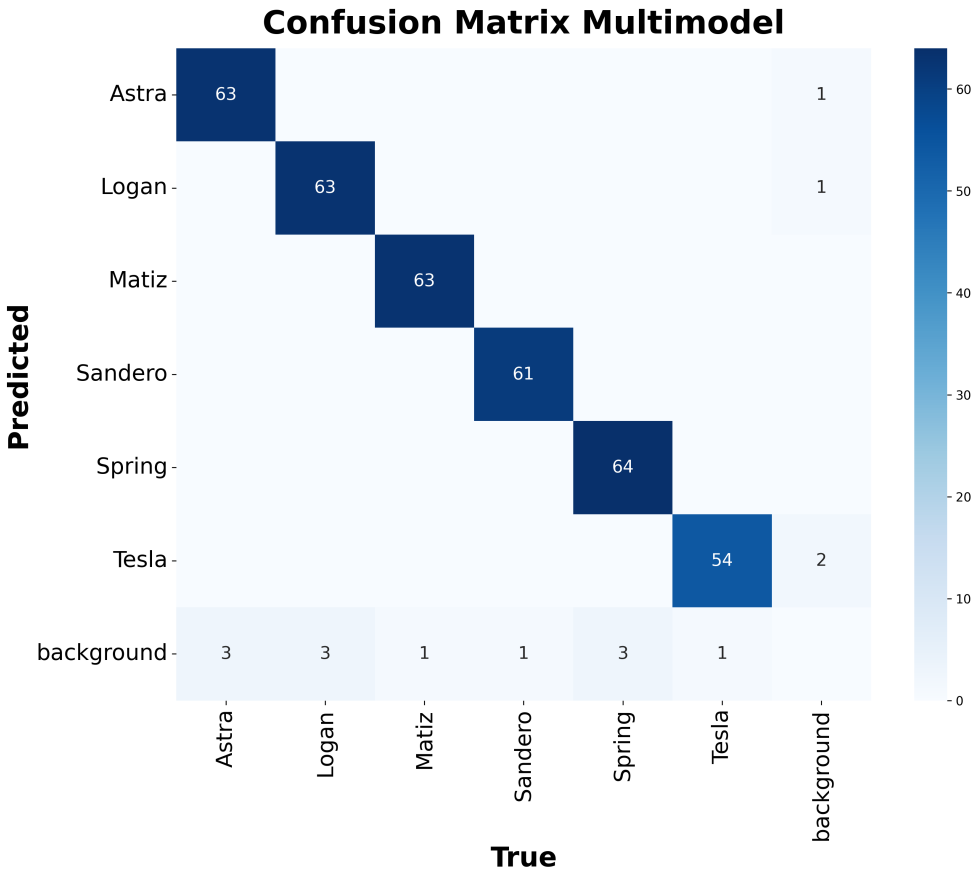

FIGURE 10. Confusion matrix for 6 vehicle classes + background for the new cascading multimodel on the initial database.

yields the confusion matrix shown in Fig. 10. A strong performance in interclass distinction was maintained, with no incorrect classifications across the six vehicle types, as highlighted by the dominant diagonal values.

Furthermore, there was a significant reduction in incorrect background predictions, with only 4 instances in which a car was detected when it should have been background. Overall, there are 16 background-related errors, with the highest frequency in "Logan" and "Astra" and the lowest in "Matiz" and "Sandero", each with just one instance.

Fig. 11 shows examples of misclassifications for each model. In case a), the YOLOv10 uncropped model detects a "Logan" class with 83% confidence, although the true model is a "Renault" car, the "Tesla" class is correctly detected with a confidence of 97%. In case b), a Spring model is not detected by the YOLOv10 cropped model; instead, part of another car is labelled as "Logan" with 77% confidence. In case c), a "Tesla" car was not identified by the newly proposed model.

FIGURE 11. a) Wrong classification of the YOLOv10 for uncropped images, b) Wrong classification of the YOLOv10 for cropped dataset, c) Wrong classification of the new cascading multimodel architecture.

6. Discussion

The results suggest that training a YOLOv10 model on a high-resolution dataset (4000x3000 pixels) is significantly constrained by the need to rescale it to 640x640 pixels for model input. The scale reduction during both training and inference adversely affects performance by decreasing the area of interest in the image, thereby reducing the sensitivity to discriminative features such as car logos, headlight structures, small details of vehicle body structures, or key geometric features. A similar effect was examined in [4] using the YOLOv3 model, where downsampling images reduced the classifier's overall accuracy.

For the cropped dataset, the YOLO classifier's performance improves because the relative object scale within the region of interest is enlarged relative to the original image, thereby preserving more details and eliminating significant uninformative background. Using this strategy, the model focused on learning highly discriminative features for each class. The importance of identifying highly discriminative features to achieve better classification results was emphasized in [5], where a similar ROI cropping strategy was employed after localizing the key discriminative regions.

The proposed multimodel cascading architecture serves as a bridge between the first YOLOv10 model trained on high-resolution images with minimal preprocessing and considerable background noise and the second YOLOv10 model trained on cropped images with much of the background noise removed and greater attention to fine discriminative details for each vehicle class. As expected, the performances were between those of the two models. The new architecture outperforms the first YOLOv10 model by 4 cases, which are no longer background-related misclassifications, but remains below the ideal case analyzed for the second YOLOv10 model trained on cropped images. The performance improvement for the background-related problems was 20%, with a decrease from 20 to 16 cases.

6.1. Limitations. One limitation of this study stems from the YOLOv10 bounding-box approach, which does not consider precise contours defined by polygons.

Consequently, during both training and inference, the region of interest (ROI) is identified using the bounding box. Converting from polygons to rectangles introduces a significant number of background pixels that are erroneously included in the ROI, potentially reducing the accuracy of the model. These inherent limitations of

bounding boxes were discussed in [6], where oriented bounding boxes were proposed as an improvement over traditional axis-aligned bounding boxes. The sensitivity of the metric, intersection over union (IoU), depends on the chosen threshold, which determines whether a detection is classified as a true positive or a false positive. This can result in visually acceptable overlaps being penalized and counted as incorrect detections, a phenomenon that becomes evident when analyzing confusion matrices with different IoU thresholds. The drawbacks of using IoU-based evaluation metrics have been examined in [21].

Another limitation of the current study is the scale and diversity of the dataset. The image collection included only six vehicle models acquired under a limited range of conditions. Most images were captured during daylight with minimal vehicle occlusion, and none showed the back of the car. As stated in [15], to develop a robust model capable of high generalization, a large and diverse set of images is necessary for the model to identify subtle discriminative features among similar vehicle classes accurately.

6.2. Future works. Future research should explore other resolution-preserving techniques beyond cropping, which should be tested on the YOLO model. Some mechanisms to consider include reconstruction-aware and fine-grained enhancement techniques that aim to minimize the impact of the trade-off between computational efficiency and resolution preservation on the performance metrics of the model. This approach is detailed in [27], where the problem is addressed by replacing the downsampling module in the backbone of the model, leading to a significant improvement in detecting small regions of interest.

Another direction is to use segmentation instead of current bounding-box annotation. This change would require a new architecture, such as Mask R-CNN, Cascade Mask R-CNN, YOLACT, or YOLOv8-seg, which better aligns with the polygon annotations available in this dataset. The drawback of this approach is the significant increase in the computational cost required for both training and inference.

7. Conclusion

In this study, three vehicle classification detection strategies based on the YOLOv10 framework are examined. In the first approach, the YOLOv10 model was trained on a set of images with minimal preprocessing. The model shows strong results in interclass differentiation, but in complex scenes, background classification problems arise. The second YOLOv10 model is trained on the same image set, however, with preprocessing that crops the image around the vehicle, thereby reducing the background noise. This significantly alleviated the background classification issue, resulting in only 4 misclassifications. Although this method yields nearly perfect results, it presupposes an ideal situation in which images are cropped to the region of interest or focused on the vehicle, with minimal background noise. To utilize both models, a new cascading multimodal architecture was proposed. The experimental results demonstrate an improved classification performance with the new architecture compared to the initial YOLOv10 trained on uncropped images, although it still falls short of

the ideal performance in the cropped scenario. This approach indicates that a cascading multimodal strategy can be a promising solution for vehicle classification in high-resolution real-world images.

References

[1] K. Aminiyeganeh, R.W.L. Coutinho, A. Boukerche, IoT video analytics for surveillance-based systems in smart cities, *Computer Communication* **224** (2024), 95–105.

[2] F. Arena, G. Pau, A. Severino, An Overview on the Current Status and Future Perspectives of Smart Cars, *Infrastructures* **5** (2020), no. 53.

[3] D. Avianto, A. Harjoko, Afiahayati, CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning, *Journal of Imaging* **8** (2022), 293.

[4] I. Bouderbal, A. Amamra, M.A. Benatia, How Would Image Down-Sampling and Compression Impact Object Detection in the Context of Self-driving Vehicles?, *Proceedings of Advances in Computing Systems and Applications*, Lecture Notes in Networks and Systems 199, Springer (2020), 25–37.

[5] W. Chen, S. Ran, T. Wang, L. Cao, Learning How to Zoom In: Weakly Supervised ROI-Based-DAM for Fine-Grained Visual Classification, *Proceedings of Artificial Neural Networks and Machine Learning – ICANN 2021*, Lecture Notes in Computer Science 12892, Springer (2021), 118–130.

[6] Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, C. Yang, PIoU Loss: Towards Accurate Oriented Object Detection in Complex Environments, *Proceedings of Computer Vision – ECCV 2020*, Lecture Notes in Computer Science 12350, Springer (2020), 195–211.

[7] CRAFT LAW FIRM, Autonomous Vehicle Accidents: NHTSA Crash Data (2019-2025). Available: https://www.craftlawfirm.com/autonomous-vehicle-accidents-2019-2024-crash-data/ [Accessed: Jan. 5, 2026]

[8] A. Elhanashi, P. Dini, S. Saponara, Q. Zheng, Integration of Deep Learning into the IoT: A Survey of Techniques and Challenges for Real-World Applications, *Electronics* **12** (2023), 4925.

[9] M. Geisslinger, F. Poszler, M. Lienkamp, An ethical trajectory planning algorithm for autonomous vehicles. *Nature Machine Intelligence* **5** (2023) 137–144.

[10] N.J. Goodall, Machine Ethics and Automated Vehicles. In: *Meyer, G., Beiker, S. (eds) Road Vehicle Automation*, Lecture Notes in Mobility, Springer (2014), 93–102.

[11] P. Hurtik, V. Molek, J. Hula, M. Vajgl, P. Vlasanek, T. Nejezchleba, Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3, *Neural Computing and Applications* **34** (2022), 8275–8290.

[12] A. Karnati, D. Mehta, K.S. Manu, Artificial Intelligence in Self Driving Cars: Applications, Implications and Challenges, *Ushus Journal of Business Management* **21** (2022), no. 4, 1–28.

[13] D. Kasraian, S. Raghav, B. Yusuf, E.J. Miller, A longitudinal analysis of travel demand and its determinants in the Greater Toronto-Hamilton Area, *Environment and Planning B: Urban Analytics and City Science* **49** (2022), no.8, 2230–2249.

[14] U. Khadam, P. Davidsson, R. Spalazzese, A systematic literature review on AI in IoT systems: Tasks, applications, and deployment, *Internet of Things* **34** (2025), 101779.

[15] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D Object Representations for Fine-Grained Categorization, *Proceedings of the IEEE International Conference on Computer Vision ICCV Workshops* (2013), 554–561.

[16] S. Ma, J.J. Yang, M.G. Chorzepa, C. Morris, S.S. Kim, S.A. Durham, Composite Deep Learning Architecture for Vehicle Classification Using Vision Transformers and Wheel Position Features, *SN Computer Science* **5** (2024), 230.

[17] J. Maurício, I. Domingues, J. Bernardino, Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review, *Applied Sciences* **13** (2023), 5521.

[18] N. Al Mudawi, A.M. Qureshi, M. Abdelhaq, A. Alshahrani, A. Alazeb, M. Alonazi, A. Algarni, Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences, *Sustainability* **15** (2023), 14597.

[19] A.A. Musa, S.I. Malami, F. Alanazi, W. Ounaies, M. Alshammari, S.I. Haruna, Sustainable Traffic Management for Smart Cities Using Internet-of-Things-Oriented Intelligent Transportation Systems (ITS): Challenges and Recommendations, *Sustainability* **15** (2023), 9859.
[20] V.Q. Nghiem, H.H. Nguyen, M.S. Hoang, LEAF-YOLO: Lightweight Edge-Real-Time Small Object Detection on Aerial Imagery, *Intelligent Systems with Applications* **25** (2025), 200484.
[21] R. Padilla, W.L. Passos, T.L.B. Dias, S.L. Netto, E.A.B. da Silva, A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit, *Electronics* **10** (2021), 279.
[22] S. Sowmiya, S. Jayasri, A.F. Thahamina, M. Srimathi, S. Raghavan, AI and IoT Integration for Next-Generation Smart Cities, *International Journal of Research and Scientific Innovation* **13** (2026), no. 2 1011–1020.
[23] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, Yolov10: Real-time end-to-end object detection, *Advances in Neural Information Processing Systems* **37** (2024), 107984–108011.
[24] M. Wang, N. Debbage, Urban morphology and traffic congestion: Longitudinal evidence from US cities, *Computers, Environment and Urban Systems* **89** (2021), 101676.
[25] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, X. Guo, Real-Time Vehicle Detection Based on Improved YOLO v5, *Sustainability* **14** (2022), 12274.
[26] X. Zhao, Y. Xia, W. Zhang, C. Zheng, Z. Zhang, YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection, *Remote Sensing* **15** (2023), 3778.
[27] W. Zheng, B. Xiong, J. Chen, Q. Ou, L. Yu, A Texture Reconstructive Downsampling for Multi-Scale Object Detection in UAV Remote-Sensing Images, *Sensors* 25 (2025), 1569.

(Andrei Gabriel Nascu) Department of Computer Science, University of Craiova, 13 A.I. Cuza Street, Craiova, 200585, Romania
E-mail address: andrei.nascu@edu.ucv.ro

(Daniel-Gheorghe Gagiu, Dan Selişteanu) Department of Automatic Control and Electronics, University of Craiova, 13 A.I. Cuza Street, Craiova, 200585, Romania
E-mail address: daniel.gagiu@edu.ucv.ro, dan.selisteanu@edu.ucv.ro