smart cities

MDPI

Article

Comparative Analysis of Traffic Detection Using Deep Learning: A Case Study in Debrecen

João Porto [1,*], Pedro Sampaio [1], Peter Szemes [2], Hemerson Pistori [1,†] and Jozsef Menyhart [2]

[1] Inovisão Department, Universidade Católica Dom Bosco, Campo Grande 79117-900, MS, Brazil; ra186488@ucdb.br (P.S.)
[2] Department of Vehicles Engineering, Faculty of Engineering, Vehicles and Mecatronics Institute, University of Debrecen, 4032 Debrecen, Hungary; szemespeter@eng.unideb.hu (P.S.); jozsef.menyhart@eng.unideb.hu (J.M.)
* Correspondence: jvaporto@gmail.com
† Deceased author.

Highlights

What are the main findings?

- Introduction of DebStreet, a dataset from Debrecen that improves regional representation for urban vehicle detection models.
- Implementation of a three-stage experimental framework showing the impact of regional data in state-of-the-art model performance.

What is the implication of the main finding?

- Regional datasets and standardized protocols improve the development of adaptable and context-aware detection systems.
- The study advances smart city traffic monitoring, offering tools for sustainable urban mobility and improved traffic management.

Abstract

This study evaluates deep learning models for vehicle detection in urban environments, focusing on the integration of regional data and standardized evaluation protocols. A central contribution is the creation of DebStreet, a novel dataset that captures images from a specific urban setting under varying weather conditions, providing regionally representative information for model development and evaluation. Using DebStreet, four state-of-the-art architectures were assessed: Faster R-CNN, YOLOv8, DETR, and Side-Aware Boundary Localization (SABL). Notably, SABL and YOLOv8 demonstrated superior precision and robustness across diverse scenarios, while DETR showed significant improvements with extended training and increased data volume. Faster R-CNN also proved competitive when carefully optimized. These findings underscore how the combination of regionally representative datasets with consistent evaluation methodologies enables the development of more effective, adaptable, and context-aware vehicle detection systems, contributing valuable insights for advancing intelligent urban mobility solutions.

Keywords: machine learning; artificial intelligence; vehicle detection; urban traffic monitoring; smart transportation and mobility

Academic Editor: Pierluigi Siano

Received: 22 April 2025
Revised: 17 June 2025
Accepted: 19 June 2025
Published: 24 June 2025

Citation: Porto, J.; Sampaio, P.; Szemes, P.; Pistori, H.; Menyhart, J. Comparative Analysis of Traffic Detection Using Deep Learning: A Case Study in Debrecen. *Smart Cities* **2025**, *8*, 103. https://doi.org/10.3390/smartcities8040103

1. Introduction

The concern regarding air quality in major metropolitan areas has intensified globally, particularly in light of the significant increase in the number of motor vehicles navigating

urban streets over the past decade, which account for more than 20% of global carbon dioxide emissions. In response to this alarming scenario, various nations have invested in research projects and public policies aimed at mitigating environmental impacts and ensuring a more sustainable future for upcoming generations. A notable example is the case of the Indian government, which has implemented proactive measures such as banning the use of two-stroke and dual-stroke engines in motorcycles widely used by the population, seeking to significantly reduce atmospheric pollutant levels [1,2].

In the European context, these concerns become even more pronounced, given the region's crucial role in the global landscape of road vehicle production. Hungary, standing out as one of the key automotive manufacturing hubs in Eastern Europe, has played a strategic role in generating knowledge and promoting advancements in the sector, considering economic, social, and ecological aspects. The country has directed substantial investments into the field of electromobility, aiming to understand and boost the positive impacts of the transition to electric vehicles in these spheres, establishing itself as a reference for sustainable innovation in the automotive industry [3,4].

To enable the control and automated estimation of vehicle incidence on public roads, the use of machine learning and deep learning technologies has gained prominence in recent years, driven by the need for intelligent solutions in urban traffic management. These technologies stand out for their ability to process large volumes of data and for the flexibility of their architectures, which can be adapted to various applications. Deep learning models can be configured both to detect and count vehicles and to support intelligent parking systems and traffic monitoring. Their scalability allows implementation in different contexts, from small towns to large metropolitan areas, contributing to the mitigation of issues such as congestion and pollutant gas emissions. This potential has stimulated investments and research, solidifying these technologies as essential for addressing urban mobility challenges in a sustainable manner [5,6].

Due to the remarkable flexibility offered by deep learning and the continuous development of this field, there is a significant heterogeneity in the architectures proposed to solve the problem of intelligent urban mobility. Studies conducted by Almeida et al. [7] and Songire et al. [8] highlight the high potential of neural networks from the YOLO Family in addressing this complex management challenge. Similarly, yet from a distinct perspective, Ahmed et al. [9] employed classical neural networks focused on regions of interest to tackle the same issue, demonstrating the diversity of possible approaches to this problem. Adopting a more progressive perspective in the field of intelligent vehicle management, several authors have explored approaches centered around attention mechanisms. These strategies range from adaptations of networks that use such mechanisms locally, as observed in the studies by Yang et al. [10] and Tolba and Kamal [11], to applications where attention mechanisms play a central role, as demonstrated by Driessen et al. [12] and Patil et al. [13].

In this context, the challenges and constant evolution of the research field become evident, with intelligent traffic management being one of the main pillars for the development of smart cities. This approach aims to enhance urban livability by reducing congestion and more effectively controlling the environmental impacts caused by motor vehicles, as discussed by [14,15] in their studies on the application of the Internet of Things and intelligent traffic management systems in urban environments.

Considering the previously discussed aspects of the field and its significant impact, as well as the contributions of [16], which emphasize the positive effects of regionally representative data on the performance and applicability of machine learning models, it becomes evident that, despite remarkable advances in the development of such models, there remains a notable lack of localized information. This gap compels deep learning and machine learning solutions designed for the future of smart cities to optimize their

performance based on generic datasets, requiring them to extrapolate learned patterns to the specific contexts in which these tools will ultimately be applied.

Thus, this work seeks to advance the field of intelligent urban mobility by emphasizing the importance of contextual and site-specific information. To this end, the DebStreet dataset was created and introduced, a new image collection composed of photographs captured in the urban environment of Debrecen, Hungary, specifically at the intersection of Bólyai and Thomas Mann Streets, under varying weather conditions. This dataset provides essential regional information that enhances the development and evaluation of detection models, making them more sensitive and better adapted to the local context.

Additionally, a comparative study was conducted involving state-of-the-art architectures for vehicle detection in urban environments, using standardized evaluation metrics that ensure the reproducibility of results and guide future research in similar scenarios. Finally, the performance of these networks was analyzed in both in-domain and cross-domain training contexts, offering a comprehensive perspective on the potential improvements enabled by training strategies that incorporate strongly regionally representative data, thus supporting the development of more effective and adaptable intelligent mobility systems.

2. Materials and Methods

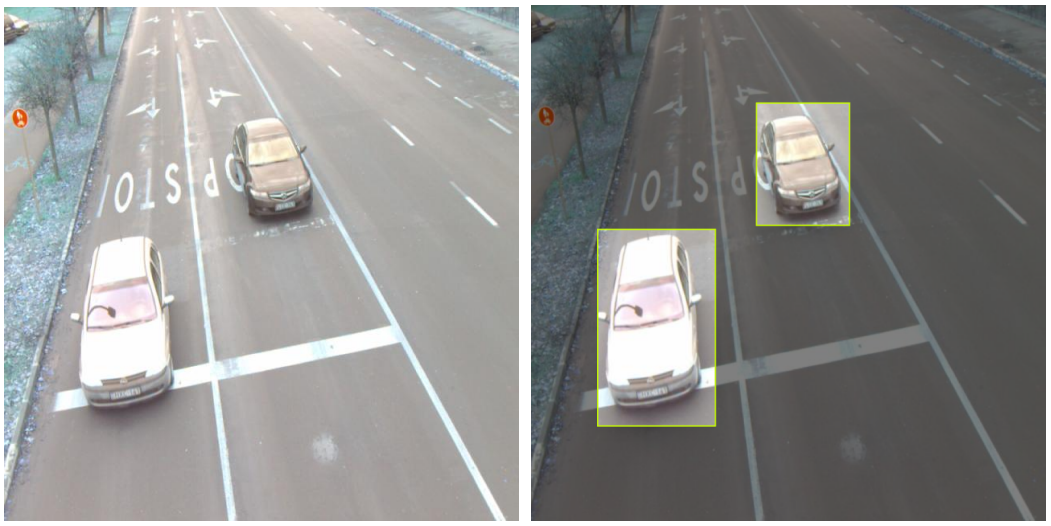
2.1. Dataset

To carry out the proposed experiments, two distinct datasets were employed. The first, used during the weight pre-training stage, was selected with the aim of adapting the investigated approaches to the general context of vehicle detection, without specifically addressing the urban scenario of the city of Debrecen. For this purpose, the annotated image dataset UA-DETRAC [17] was chosen.

The selection of this dataset is justified by its broad diversity and the high quality of the annotated objects, with a particular focus on urban environments, making it a well-established reference for the evaluation of models in the vehicle detection domain. Furthermore, as highlighted by Liang et al. [18] in their analysis of the most relevant datasets in the field, the use of a comprehensive and representative dataset such as UA-DETRAC is of great significance for experiments of this nature, being widely recognized in the literature as a reliable benchmark. In this study, a random sample of 10,000 images was selected from this dataset to serve as the training base for the models in the initial experimental phase.

Based on previously trained and validated weights, to ensure that the employed architectures possessed the necessary foundation for analyzing information within the proposed context, a second set of images was utilized. This set corresponds to a novel dataset, introduced for the first time in this article, named DebStreet (dataset available at: https://universe.roboflow.com/joao-vitor-de-andrade-porto/debstreet, accessed on 18 June 2025), which captures information from the location under controlled climatic and lighting conditions, inspired by the way the Caltech 1999 [19] and 2001 [20] datasets were constructed.

This set comprises 682 images, totaling 3256 annotated objects of interest, all labeled with the class “Vehicle”, as illustrated in the example shown in Figure 1, without the application of any data augmentation techniques. The images were captured in the urban environment of Debrecen, Hungary, using a camera positioned at the height of a traffic light, enabling the visual recording of vehicles both stationary and in motion at the intersection of Bólyai and Thomas Mann Streets. This location is widely recognized for its high volume of private vehicle traffic and the vehicular diversity observed, due to the presence of two intersecting roads and the circulation of trams.



(**a**) Sample image. (**b**) Annotated sample.

Figure 1. Visual representation of how the image is presented in the dataset (**a**) and how it was annotated (**b**).

2.2. Architectures

For the initial weight training with the UA-DETRAC dataset, four distinct architectures were selected for performance comparison, chosen based on their structural characteristics. The categories considered were the YOLO Family, Region Proposal Networks, attention mechanisms, and Vision Transformers. As a representative of the YOLO Family, the YOLOv8 network [21] was selected due to its high performance and widespread popularity in recent studies in the field, as discussed in the Introduction. For the Region Proposal Networks, Faster R-CNN [22] was chosen, standing out for its relevance in deep learning-based object detection and its robustness in handling diverse data.

In the context of attention mechanisms, the Side-Aware Boundary Localization (SABL) architecture [23] was selected, considering the positive impact of this mechanism, as evidenced in its original study. Finally, to represent Transformer-based networks, the Detection Transformer (DETR) [24] was chosen due to its unique encoder–decoder architecture for object detection, which significantly simplifies the overall process. These selections reflect the pursuit of a comprehensive analysis of the leading contemporary approaches in object detection.

2.3. Comparative Metrics

In order to ensure the comparability of the presented results, and based on the considerations by [25] regarding the importance of evaluation metrics, a set of nine comparative metrics was selected. The first six are directly associated with the classification and detection processes widely used in the literature, encompassing the calculation of general Mean Average Precision (mAP), as well as the specific Intersection over Union values of 50% (mAP50) and 75% (mAP75), in addition to the traditional classification metrics: Precision, Recall, and Fscore.

To obtain the values of the adopted metrics, at the end of each test procedure, the results were analyzed based on the correct and incorrect detections as well as classifications. The corresponding metrics were then calculated according to the equations presented below.

For the three most commonly used metrics, Precision, Recall, and Fscore, the reported and analyzed values were obtained from Equations (1), (2), and (3), respectively.

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \tag{1}$$

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{2}$$

$$\text{Fscore} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

In these equations, N represents the number of classes; TP_i denotes the number of true positives, or correctly predicted instances of class i; FP_i represents false positives, or instances incorrectly predicted as class i; and FN_i signifies false negatives, or instances of class i that were incorrectly predicted as another class.

Regarding the calculation of Mean Average Precision at different levels of Intersection over Union (IoU), Equations (4), (5), and (6) were used to obtain the values of mAP, mAP50, and mAP75, respectively.

In these equations, N denotes the total number of object classes, and $\text{AP}_i^{(\text{IoU}_k)}$ represents the Average Precision for class i, which is computed at a specific Intersection over Union (IoU) threshold, denoted as IoU_k. The general Mean Average Precision (mAP) is obtained by averaging the AP values across K different IoU thresholds, which typically range from 0.50 to 0.95 in increments of 0.05, following the COCO evaluation protocol. The metrics mAP@50 and mAP@75 represent the mean precision calculated using only predictions where the IoU is greater than or equal to 0.50 and 0.75, respectively. These metrics reflect the model's performance under increasingly stringent localization criteria.

$$\text{mAP} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N}\text{AP}_i^{(\text{IoU}_k)}\right) \tag{4}$$

$$\text{mAP@50} = \frac{1}{N}\sum_{i=1}^{N}\text{AP}_i^{(\text{IoU}\geq 0.50)} \tag{5}$$

$$\text{mAP@75} = \frac{1}{N}\sum_{i=1}^{N}\text{AP}_i^{(\text{IoU}\geq 0.75)} \tag{6}$$

Finally, three additional metrics specific to this experiment were selected. Although traditionally used in regression studies, these metrics, as discussed by [26] in their research on green apples, can contribute to assessing the robustness and the model's ability to fit the problem, allowing for a more comprehensive performance analysis. Thus, the metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Pearson's correlation coefficient (r) were adopted.

For the calculation of these specific experimental metrics, Equations (7)–(9) were employed, where n represents the total number of samples, y_i corresponds to the actual value of the i-th sample, $\hat{y}_i$ denotes the predicted value for the i-th sample, $\bar{y}$ indicates the mean of the actual values y across all samples, and $\bar{\hat{y}}$ represents the mean of the predicted values $\hat{y}$ within the same set.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \quad (9)$$

2.4. Experimental Setup

To determine the most suitable approach for vehicle detection in an urban context, it became essential to establish quantitative criteria for comparing the methods analyzed. To this end, the study adopted two types of analysis for each experiment. The first focuses on evaluating each network's ability to classify and identify objects of interest, using metrics such as Precision, Recall, Fscore, Mean Average Precision (mAP), mAP at 0.75 IoU (mAP75), and mAP at 0.5 IoU (mAP50). These metrics were selected to standardize comparisons with previous studies in the literature, thereby facilitating consistent and meaningful future analyses.

The second analysis focused more specifically on the problem of quantifying the number of vehicles passing through the roadway. For this purpose, a distinct set of metrics was adopted, aimed at vehicle counting. The metrics selected for this analysis were the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Pearson correlation coefficient (r), which allowed for a more precise evaluation of the architectures within the specific context of the problem under study.

As previously presented, the main experiment of this study, which used both datasets, was carried out in three stages, with metrics being collected and evaluated at each stage. The first stage consisted of training the architectures with the goal of ensuring generalization in data prediction, focusing on the urban vehicle traffic scenario. For this stage, the UA-DETRAC dataset underwent a random sampling process without replacement, resulting in the formation of 10 distinct subsets, each containing exactly 10% of the total images. These subsets were used in a k-fold cross-validation procedure, conducted with 10 folds. In each iteration, one of the subsets was designated as the test set, while the remaining subsets were allocated for training and validation in an 80:20 ratio. The metrics previously discussed were collected and stored at the end of each fold.

The adopted cross-validation strategy ensured that each of the 10 subsets was used exactly once as a test set, providing a comprehensive assessment of the model's performance. Furthermore, the strict separation between the training, validation, and test sets prevented any data contamination. In each iteration, the model's weights were reset, ensuring that the process started from scratch. This approach reinforces the reliability of the obtained results, enabling a robust statistical analysis of the architectures' performance. To ensure the standardization of the experiments, the hyperparameters learning rate, number of epochs, batch size, and patience were kept constant across all executions, assuming the values N, M, O, and P, respectively.

In order to determine these values, the image set was shuffled and five executions were performed, varying one hyperparameter at a time while keeping the others fixed. This procedure was repeated for each hyperparameter in order to identify the optimal combination for the conducted experiment. At the end of the executions, the loss curves during training and validation were analyzed; the best configuration was identified based on the region of the graph where the curve begins to stabilize, although it still shows a slight downward trend. Thus, the values of N, M, O, and P were defined in this study as 0.001, 200, 16, and 10 (5%), respectively, for all experimental runs performed.

To perform the statistical comparison of the stored quality metrics, the obtained values were subjected to three distinct analyses: analysis of the mean values and standard deviation, construction and analysis of boxplots, and execution of the analysis of variance (ANOVA) test with a post hoc Tukey test, adopting a 5% significance level. Through this robust statistical analysis, it was possible not only to identify the differences between the

results presented by the architectures, but also to assess the relevance of these differences and the likelihood of their existence in an average sample outside the experimental control data, in the context of real-world application scenarios.

In the second stage, the weights from each fold of each architecture were reused for testing and metric collection; however, this time, the DebStreet dataset was used in place of the test set from each fold. The same metrics and statistical analyses were applied with the aim of evaluating the performance of the weights learned on a general dataset when applied to a new problem, using a cross-domain approach, without fine-tuning.

Finally, the third stage consisted of reapplying the 10-fold cross-validation process to all architectures, using the pre-trained weights obtained in Stage 1 of the experiment, now combined with the specific DebStreet dataset for classification, detection, and counting tasks within the proposed context. Thus, fine-tuning of the previously trained weights was performed to adapt each architecture to the new application domain. The collection and analysis of the same metrics from the previous stages were maintained, characterizing the cross-domain configuration with fine-tuning. This experiment aimed not only to identify the most suitable network for solving the problem but also to demonstrate the generalization and adaptability capacity of each architecture when adjusted to a new sample set.

All experiments were conducted over a five-day period using an NVIDIA A2000 graphics card with 12 GB of dedicated GDDR6 memory. The architectures were implemented using the PyTorch library version 2.5.0, while the Side-Aware Boundary Localization model was developed based on the MMDetection framework [27] accessed on 15 March 2025 (the implementations used in this experiment are publicly available at the following repositories: compara_detectores_torch (https://github.com/Inovisao/compara_detectores_torch), author's implementation of the Faster R-CNN, DETR, and YOLOv8 networks using PyTorch; and detectores_json_k_dobras (https://github.com/Inovisao/detectores_json_k_dobras), MMDetection wrapper for statistical data generation and experiment configuration).

3. Results and Discussion

The three stages of the process were executed sequentially, ensuring coherence and continuity in the analysis. The results obtained in each stage were duly reported and are discussed in Sections 3.1, 3.2, and 3.3, which detail the procedures adopted for evaluating the model's generalization, transferability, and adaptability, respectively. This approach enabled a systematic and structured analysis, providing a better understanding of the model's performance under different scenarios and conditions, which were previously described in the Materials and Methods Section.

3.1. Same-Domain Generalization Stage

Regarding the metrics from Stage 1 of the experiment, the attention mechanisms category stood out significantly through the SABL architecture. The analysis of Table 1 reveals that the values highlighted in bold, representing the best results for each column, belong exclusively to the SABL architecture, with all values exceeding 80% (0.80) and exhibiting an extremely low standard deviation. This indicates the uniformity and consistency of the architecture's performance on the trained dataset. In contrast, although DETR did not always yield the lowest values, such as in the case of the mAP50 metric, where Faster R-CNN and YOLOv8 obtained lower values, the Transformer-based architecture exhibited the highest standard deviations among all analyzed architectures. This suggests significant instability and unpredictability, directly contrasting with the attention mechanisms category.

Table 1. Mean values and standard deviations of each architecture in the standardized analysis relative to mAP values. The values highlighted in bold correspond to the highest for each analyzed variable, while the lowest standard deviation may indicate greater relevance.

Architecture	mAP	mAP50	mAP75
YOLOV8	0.629 ± 0.003	0.727 ± 0.003	0.709 ± 0.003
Faster	0.487 ± 0.008	0.627 ± 0.006	0.590 ± 0.006
Detr	0.493 ± 0.125	0.731 ± 0.157	0.554 ± 0.162
Sabl	**0.848 ± 0.005**	**0.981 ± 0.006**	**0.965 ± 0.006**

Similarly to the metrics related to mAP, the values obtained for Precision, Recall, and Fscore also highlight the Side-Aware Boundary Localization architecture, as presented in Table 2. This table reveals a general reduction in the metrics for the highlighted architecture, with all three metrics close to 78% (0.78), yet still superior, and exhibiting a low standard deviation. This reinforces the scores observed in the mAP analysis. On the other hand, the Transformer-based category remained the most inconsistent, despite a considerable reduction in the standard deviation for all three metrics.

Table 2. In the standardized analysis of Precision, Recall, and Fscore metrics, the mean values and standard deviations of each architecture were calculated. The highest values for each analyzed variable are highlighted in bold, while the lowest standard deviation may indicate greater relevance.

Architecture	Precision	Recall	Fscore
YOLOV8	0.719 ± 0.004	0.744 ± 0.001	0.731 ± 0.002
Faster	0.589 ± 0.007	0.633 ± 0.007	0.608 ± 0.005
Detr	0.512 ± 0.095	0.651 ± 0.051	0.561 ± 0.079
Sabl	**0.781 ± 0.005**	**0.790 ± 0.006**	**0.785 ± 0.005**

The factors observed in the analysis of standardized variables serve as important indicators of the applicability of architectures with attention mechanisms in problems characterized by high heterogeneity of information within the same class. The studies of [28,29] support these conclusions by examining the impact of such mechanisms on the solutions proposed by the authors. Likewise, the limitations identified in the works of [30,31] are also reflected in the data obtained from the standardized variables of the experiment.

Regarding the analysis of specialized metrics, as reported in Table 3, the previously observed trend of SABL outperforming all other architectures across all metrics does not hold in this context. For the error-related metrics, the YOLOv8 network achieved the lowest values for both MAE and RMSE, indicating that it commits fewer errors in terms of object counting compared to the other architectures. As for the correlation coefficient r, all four approaches exhibit a strong positive correlation, suggesting that their counting behavior closely aligns with the actual values. Notably, the SABL architecture, despite not being the most accurate in terms of error incidence, demonstrates the highest positive correlation with the ground truth, reinforcing its reliability in count estimation.

The trend identified in the other metrics was maintained. In this context, the attention mechanism category stood out once again, achieving the lowest error values in MAE and RMSE, along with a high degree of correlation, as indicated by the r metric, and low standard deviations across all three metrics. Similarly, the Transformer-based category continued to exhibit significant variability in the data. However, in the case of error metrics, its performance differed from that observed in the generalized metrics, as it consistently yielded qualitatively inferior values compared to the other approaches. Regarding correla-

tion, DETR obtained higher values than other methods, such as Faster R-CNN, though it did not surpass the SABL network.

Table 3. For the specific analysis of the studied problem, the mean values and standard deviations of each architecture are presented. The highest values for each variable are highlighted in bold, while lower variability, indicated by a reduced standard deviation, may suggest greater relevance.

Architecture	MAE	RMSE	r
YOLOV8	**0.210** ± **0.023**	**0.652** ± **0.101**	0.985 ± 0.003
Faster	0.770 ± 0.051	1.510 ± 0.116	0.907 ± 0.007
Detr	1.381 ± 0.361	2.051 ± 0.418	0.884 ± 0.021
Sabl	0.236 ± 0.017	0.665 ± 0.067	**0.990** ± **0.002**

For the second analysis outlined in the methodology, aimed at examining the distribution of fold values in greater detail graphically through boxplots, Figure 2 illustrates these distributions along with their interquartile ranges and medians for each metric, relating the studied approaches.

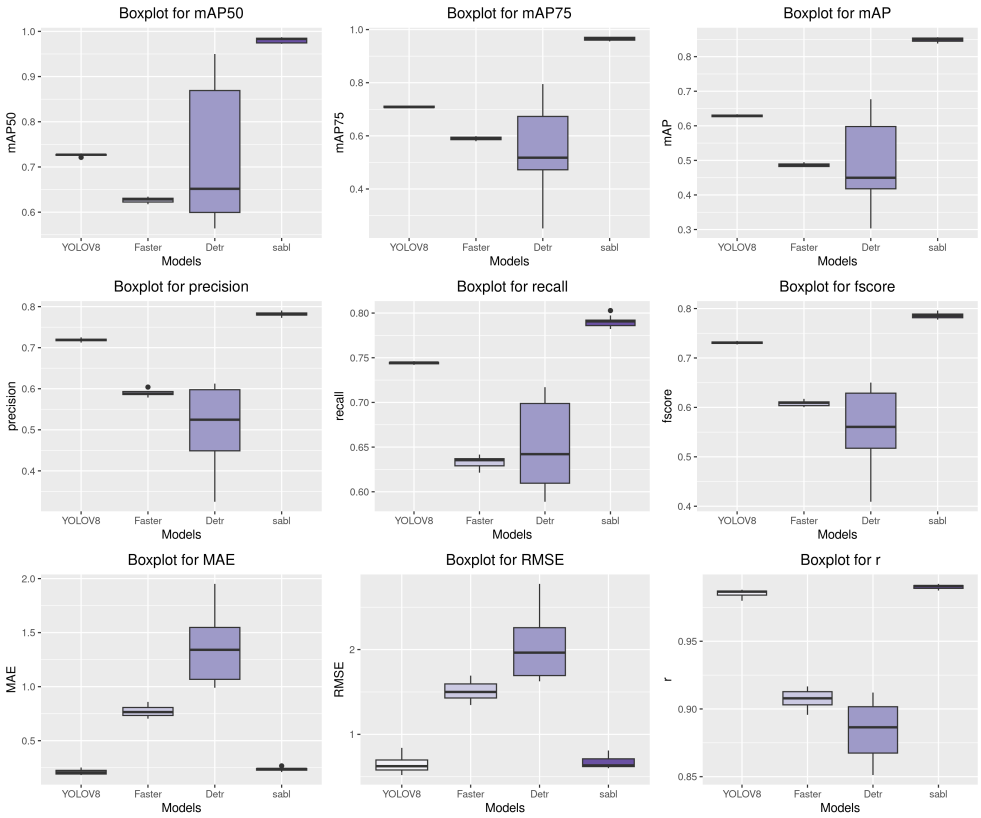


Figure 2. The graph displays the boxplots for each metric in relation to its respective architecture. The first two rows correspond to the standardized metrics, while the last row presents the three metrics specific to the experiment.

Upon analyzing the boxplot, it is observed that the SABL neural network once again appears to outperform the others, particularly in the standardized metrics, where it achieved the best performance and a noticeable difference compared to the other approaches. However, in the specific counting metrics, YOLOv8 demonstrated superior performance in terms of MAE and RMSE, while SABL maintained a similar performance in the r metric.

These proportions and values were also corroborated by the analysis of variance (ANOVA), which indicated p-values significantly lower than 0.05 for all metrics, ranging from 0.0087 to 0.00034, thus evidencing the existence of potential statistical differences

among the approaches. The Tukey multiple comparisons test enabled the identification of the specific pairs of approaches for which these differences occurred, confirming the prominence of SABL in the standardized detection metrics, for which it presented *p*-values below 0.0074. However, for the specific counting metrics, no evidence of statistical difference was observed between SABL and YOLOv8, since the comparison between them resulted in a *p*-value of 0.0724, exceeding the previously established significance level.

This behavior may be related to the ability of both architectures to learn relevant information. While the SABL architecture addresses the challenge of heterogeneity and large information volume through its attention mechanism [29], the YOLOv8 neural network focuses on optimizing information flow, making it more robust to diverse scenarios, as indicated by [32].

3.2. Cross-Domain Transferability Stage

When analyzing the transferability of the models, the Faster R-CNN neural network exhibited the most notable performance, despite having achieved poor results in the previous stage and not being statistically distinguishable from the worst-performing models. In this stage, which aims to assess the ability of architectures to transfer learned knowledge to new samples, this network achieved the highest values in all three mAP metrics, as well as in the Recall and Fscore metrics, as highlighted in bold in Tables 4 and 5.

Table 4. Mean and standard deviation of the mAP metrics, with boldface highlighting the most representative value in each column.

Architecture	mAP	mAP50	mAP75
YOLOV8	0.270 ± 0.033	0.546 ± 0.063	0.212 ± 0.035
Faster	**0.412 ± 0.015**	**0.835 ± 0.022**	**0.325 ± 0.034**
Detr	0.103 ± 0.055	0.287 ± 0.156	0.039 ± 0.023
Sabl	0.379 ± 0.018	0.780 ± 0.017	0.299 ± 0.034

Table 5. Tabular representation of the mean and standard deviation obtained when computing the Precision, Recall, and Fscore metrics, with textual emphasis on the highest value in each column usind bold font.

Architecture	Precision	Recall	Fscore
YOLOV8	**0.925 ± 0.041**	0.559 ± 0.069	0.693 ± 0.049
Faster	0.880 ± 0.013	**0.849 ± 0.023**	**0.864 ± 0.012**
Detr	0.417 ± 0.188	0.467 ± 0.187	0.437 ± 0.183
Sabl	0.853 ± 0.040	0.794 ± 0.017	0.822 ± 0.017

Observing Table 5, it is evident that the YOLOv8 approach outperformed the Faster R-CNN network in terms of Precision but fell short in the other metrics. This suggests that, although YOLOv8 is more effective in reducing false positives, it exhibits a higher number of false negatives. Consequently, despite having the lowest number of irrelevant objects classified as the target class, its overall performance was significantly impacted, especially in contexts where the false negative rate is crucial, as in the present experiment.

Similar to the behavior of Faster R-CNN, the attention-based SABL architecture exhibited a balanced performance between Precision and Recall, maintaining a consistent range. Although its values were lower than those of Faster R-CNN, SABL preserved the strong performance observed in the previous stage. This behavior can also be verified in Table 6 through the specific counting metrics, in which SABL achieved the lowest error values and the highest correlation among all networks analyzed in this stage.

Table 6. Specific average counting metrics reported along with their respective standard deviations, with indicative highlighting for the best value of each metric using bold font.

Architecture	MAE	RMSE	r
YOLOV8	2.114 ± 0.395	2.914 ± 0.465	0.669 ± 0.176
Faster	0.868 ± 0.098	1.447 ± 0.197	0.891 ± 0.033
Detr	2.488 ± 0.972	3.263 ± 0.059	0.178 ± 0.345
Sabl	**0.840** ± **0.132**	**1.282** ± **0.198**	**0.926** ± **0.022**

As indicated in Table 6, the median and the distribution ranges of the data presented in Figure 3 followed a similar pattern when comparing the values obtained by the Faster R-CNN and SABL architectures. The SABL architecture appears to show a slight advantage across the three count-specific metrics, based solely on the numerical values of the mean and standard deviation.

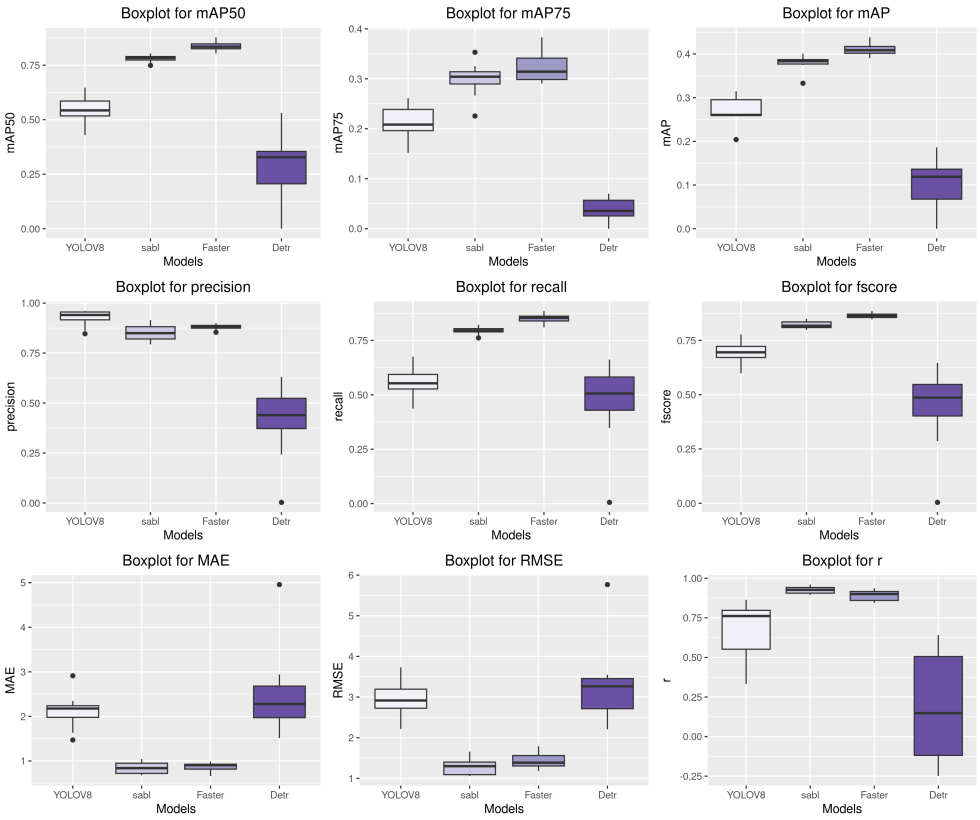


Figure 3. Boxplots representing the distributions of means and interquartile ranges for the metrics evaluated in the second stage of the experiment. The last row displays the specific counting metrics analyzed in this study, while the remaining rows present classification metrics to standardize comparisons across experiments.

The analysis of variance (ANOVA) provided relevant insights at this stage, despite the alternation between the best- and worst-performing approaches observed in the mean values presented in the tables. Although the ANOVA test indicated the possibility of statistically significant differences, with p-values ranging from 0.0234 to 0.0096, well below the 5% significance level, the Tukey post hoc test did not identify clear distinctions between the performance of the Faster R-CNN and SABL networks, as the p-values remained above 0.05 for all metrics. These results suggest that, up to the second stage, the attention-based approach exhibited the best performance, as it not only outperformed the others in the previous stage but also maintained its relevance in the scenario analyzed at this stage.

3.3. Cross-Domain Adaptability Stage

In the final stage, during the execution of fine-tuning, the standardized classification metrics did not reveal significant differences among the analyzed networks, as indicated in Tables 7 and 8, whose average values exhibited a high degree of proximity. However, it is noteworthy that, for the first time in this study, the DETR architecture achieved superior performance in one of the evaluated metrics, specifically in the mAP50 metric.

Table 7. The mean values and standard deviations of each architecture were tabulated according to the mAP metrics, with the most representative values highlighted in bold. Notably, for the first time in the experiment, the DETR architecture outperformed the others in mAP50.

Architecture	mAP	mAP50	mAP75
YOLOV8	**0.750** ± **0.010**	0.997 ± 0.003	0.903 ± 0.017
Faster	0.689 ± 0.013	0.994 ± 0.006	0.844 ± 0.018
Detr	0.710 ± 0.014	**0.998** ± **0.012**	0.863 ± 0.025
Sabl	0.745 ± 0.011	0.989 ± 0.005	**0.907** ± **0.014**

Table 8. Schematic representation of the mean values and distribution of classification metrics using standard deviation, with emphasis using bold font on the most relevant values in each column.

Architecture	Precision	Recall	Fscore
YOLOV8	0.975 ± 0.007	**0.999** ± **0.001**	0.987 ± 0.004
Faster	0.920 ± 0.008	0.997 ± 0.003	0.957 ± 0.004
Detr	0.936 ± 0.035	0.994 ± 0.002	0.964 ± 0.019
Sabl	**0.986** ± **0.008**	0.993 ± 0.004	**0.990** ± **0.004**

Analyzing the values presented in these tables, it is evident that the standard deviation was generally extremely low. This is particularly noticeable in Table 8, especially in the Recall metric for YOLOv8, where the combination of the mean value and the standard deviation is remarkably close to 100%. This result reinforces and empirically validates the ideas proposed by [33,34] regarding the impact of transfer learning and fine-tuning in optimizing models for solving real-world problems.

This proportionality and data proximity are also reflected in the mean values of the specific counting metrics analyzed in this experiment, as outlined in Table 9. This table allows for an analysis of the result distribution, with YOLOv8 once again leading in error metrics, although exhibiting a lower correlation coefficient than SABL.

Table 9. Experiment-specific metrics associated with their respective architectures, with emphasis on the most notable values for each using bold font as visual representation of it.

Architecture	MAE	RMSE	r
YOLOV8	**0.009** ± **0.014**	**0.057** ± **0.078**	0.994 ± 0.001
Faster	0.089 ± 0.034	0.316 ± 0.069	0.984 ± 0.002
Detr	0.242 ± 0.179	0.556 ± 0.304	0.981 ± 0.010
Sabl	0.041 ± 0.019	0.190 ± 0.073	**0.995** ± **0.002**

From the diagrams presented in Figure 4, it is evident that the various approaches benefited significantly from fine-tuning, leading to an overall reduction in interquartile ranges and an increase in medians compared to Figure 3 from the previous stage. Notably, the DETR architecture, despite not achieving the best absolute results, exhibited the greatest performance improvement among the evaluated approaches. Initially, in Stage 1, this network showed large variations in the range of 10%, whereas in the final stage, these variations were reduced to approximately 1%.

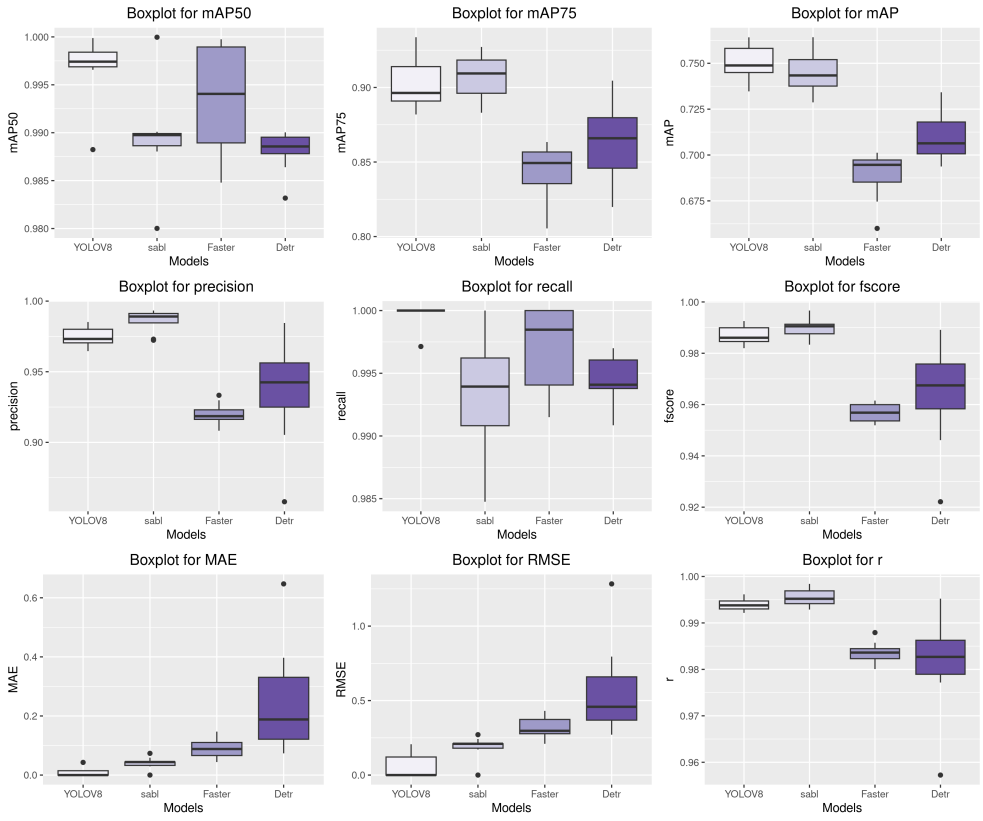


Figure 4. Distribution of values around the medians using boxplots for robust statistical analysis. Each set of four boxplots represents one of the evaluated metrics.

Regarding the ANOVA test, the similarity of the *p*-values across all metrics remained statistically significant, with a value of 0.0329, below the threshold previously established in the methodology. As illustrated in Figure 4, the similarities previously observed between SABL and YOLOv8 persisted after analyzing the values obtained in the Tukey post hoc test, resulting in a *p*-value of 0.0648. However, a notable aspect emerged: this time, DETR did not differ significantly from the other architectures, showing no statistical evidence of distinction from Faster R-CNN across several metrics, such as Precision, Recall, and Fscore, with comparative *p*-values for this pair ranging from 0.0547 to 0.0613. Additionally, in some cases, such as the Recall and mAP50 metrics, there was also no statistical evidence of distinction between DETR and SABL, with *p*-values slightly above 0.0578 but not exceeding 0.0681.

Considering the insights derived from the statistical tests, the lack of significant differences, and the average values obtained, it becomes clear that employing diverse strategies is essential to enhance model performance. These approaches made it possible to achieve high levels of Precision, mAP, and Recall, which are comparable to or even exceed those reported in related works, such as [35,36]. Both studies applied specific modifications to YOLOv8, achieving precision rates above 90 percent, a result similar to that observed in this study after the third experiment.

The analyses conducted on the performance of the architectures throughout the different stages allow for the identification of the most suitable model for each scenario, emphasizing the internal characteristics of each architecture and the benefits they offer in solving specific problems. Regarding network performance in relation to the available data and training time, the premises presented by [37,38] were empirically validated. These authors assert that as the volume of data and training time increase, attention-based networks, such as SABL and DETR, exhibit the most significant performance improvements.

This suggests that for long-term applications where continuous retraining with new data is feasible to adapt the model to real-world conditions, these two architectures are the most suitable choices.

Regarding the application of the techniques in real-world scenarios outside the control group, it is observed that the performance of the approaches significantly improves as the network training is refined with a specific focus on the environment of application. This trend became evident in the progressive increase of performance indicators, as demonstrated in the three experiments conducted, where the values, especially the Recall of YOLOv8, gradually approached 100%. Moreover, the evolution in the efficiency of the employed architectures is noteworthy. Previously, it was necessary to have computers with high computational resources; today, it is possible to run these models with satisfactory metrics and high responsiveness on devices with reduced computational capabilities. This evolution is supported by the works of [39,40], which demonstrate that the learned weights can be optimized for operation on smaller, portable devices without compromising performance.

Despite the remarkable performance, it is essential to highlight the computational resources required by each architecture. A significant consumption of both RAM and dedicated memory was observed, particularly in the case of the YOLOv8 architecture. Although it exhibited superior performance in terms of results, this network was the most computationally demanding, requiring approximately 10 GB of the dedicated GPU memory and up to 25 GB of the device's total RAM during a training period of about four and a half hours for processing its set of folds.

In contrast, the DETR architecture, although it did not achieve the best performance among the three evaluations conducted, demonstrated a more efficient computational profile. Its use of dedicated memory was equivalent to that of YOLOv8; however, its RAM consumption was substantially lower, limited to approximately 10 GB. Furthermore, the execution time was reduced to about three hours and fifteen minutes, indicating a lower computational cost compared to YOLOv8.

Thus, it becomes possible to investigate the feasibility of applying these architectures in embedded systems installed at strategic locations throughout the city, considering not only performance but also computational cost, with the aim of achieving more effective traffic control and generating data for the development of public platforms. Such platforms could utilize this information to improve citizens' daily lives, either by streamlining traffic flow or by guiding the planning and implementation of new roadways. Furthermore, this new approach to applying and processing data enables intelligent vehicle control, since it becomes feasible to identify the precise location of each vehicle, as demonstrated by the application of the YOLO network in the study by [41].

4. Conclusions

Based on the analyses conducted and the insights gathered, it can be concluded that attention mechanisms have broad applicability in vehicle identification and counting in urban environments. Regardless of whether the model is trained on a specific or a generic dataset, it is likely to achieve satisfactory performance in solving problems within this domain.

In the context of attention mechanisms, the significant impact of data volume and training time on models based on the general Transformer approach, originally proposed by [37], is evident. In this regard, the DETR architecture exhibited the most notable improvement in its metrics following fine-tuning, suggesting that, in larger-scale scenarios, it may reach the performance level observed in SABL or even surpass other architectures.

Another noteworthy aspect observed in this experiment is the behavior of the Faster R-CNN architecture. Despite being considered an older network, published nearly a decade ago, it still exhibits competitive performance in specific scenarios, such as in Stage 2. This finding underscores the importance of not only exploring new architectures for problem-solving but also investigating novel approaches to optimize and repurpose older architectures, as demonstrated in the works of [42,43], which propose variations of Faster R-CNN.

Despite the positive aspects identified and the promising prospects for future applications, it is essential to highlight the limitations observed throughout the experiment. Although the DebStreet image dataset adequately represents the studied urban scenario and yielded satisfactory results, with indications of reasonable applicability beyond controlled environments, it still lacks diversity in terms of weather conditions and significant lighting variations. This limitation may hinder the generalization capability of models trained exclusively on this dataset. To mitigate this constraint and enhance the dataset's robustness, it is recommended that future versions include samples collected during different seasons of the year and at various times of day, thereby increasing its representativeness and ensuring broader coverage of urban scenarios.

In addition to temporal and lighting variations, another aspect that can significantly improve the robustness of the dataset is the spatial diversification of the urban environments represented. While the current version of the dataset focuses primarily on intersections, future iterations could benefit from the inclusion of other urban contexts, such as major avenues and narrower pathways like curved alleyways. Expanding the scope of image acquisition to encompass a broader range of urban scenarios would increase the dataset's regional representativeness and the variability of information available for training. This is especially relevant in the context of smart cities, where intelligent systems must be capable of operating effectively across a wide variety of real-world conditions and infrastructures.

To address the limitations of the image dataset and improve its performance, the adoption of training methodologies within the field of few-shot learning emerges as a viable strategy. Although the dataset contains a relatively small number of images, their relevance and representativeness make it feasible to train models in this machine learning paradigm, potentially achieving satisfactory results. This approach is supported by the literature reviews of [44,45], which highlight few-shot learning as a promising technique for optimizing networks to operate effectively with limited datasets.

Another potential direction for future research lies in the in-depth investigation of hyperparameter tuning. As previously discussed, the architecture based on Transformer technology was among those that exhibited the greatest metric improvements throughout the experiment. This finding raises the possibility of conducting further experiments with a greater number of training epochs and a more diverse dataset. As noted earlier, this architecture significantly benefits from extended training time and increased data volume. Therefore, it is plausible to assume that in future experiments, the DETR network may achieve even better performance, potentially outperforming other evaluated approaches.

In summary, the field of vehicle detection still holds significant untapped potential, enabling the application of various techniques and architectures aimed at enhancing result accuracy and optimizing intelligent monitoring of urban roadways. This potential is further enhanced when solutions are enriched with information pertinent to the local context of deployment, as exemplified by the DebStreet dataset introduced in this article. Of particular importance is the role of the studied approaches as support tools for urban management by governmental agencies, facilitating the identification of congestion points and fostering more effective decision-making. Additionally, these tools can be employed to assess the environmental impact caused by vehicular traffic, promoting actions focused on improv-

ing urban conditions and mitigating the environmental effects associated with mobility. Such contributions may be realized through both public initiatives and partnerships with organizations capable of interpreting and utilizing the provided data.

Author Contributions: Conceptualization, J.P., P.S. (Peter Szemes), and J.M.; methodology, J.P. and H.P.; software, J.P. and P.S. (Pedro Sampaio); validation, P.S. (Peter Szemes) and H.P.; formal analysis, P.S. (Peter Szemes) and J.M.; investigation, J.P. and P.S. (Pedro Sampaio); resources, H.P. and J.M.; data curation, P.S. (Peter Szemes); writing—original draft preparation, J.P. and P.S. (Pedro Sampaio); writing—review and editing, J.P.; visualization, P.S. (Pedro Sampaio); supervision, H.P. and J.M.; project administration, H.P.; funding acquisition, H.P. and J.M. Author Hemerson Pistori passed away prior to the publication of this manuscript. All other authors have read and agreed to the published version of this manuscript.

Funding: This work has received financial support from the Dom Bosco Catholic University, the Foundation for the Support and Development of Education, Science and Technology from the State of Mato Grosso do Sul, FUNDECT, and Federal University of Pampa, UNIPAMPA. Some of the authors have been awarded with Scholarships from the the Brazilian National Council of Technological and Scientific Development, CNPq and the Coordination for the Improvement of Higher Education Personnel, CAPES.

Data Availability Statement: The dataset DebStreet can be found publically at https://universe.roboflow.com/joao-vitor-de-andrade-porto/debstreet (accessed on 18 June 2025).

Acknowledgments: We dedicate this work to the memory of Hemerson Pistori, whose invaluable support and guidance were instrumental in shaping this research. His contributions and mentorship have left a lasting impact, and his legacy continues to inspire our work. We would especially like to express our gratitude to the Department of Vehicles Engineering at the Faculty of Engineering, Vehicles and Mechatronics Institute, University of Debrecen, for their support during the research and its developments, and especially to József Menyhárt for his valuable contribution. The authors acknowledge the use of large language models such as ChatGPT (accessed on 18 June 2025) and DeepL (accessed on 18 June 2025) alongside the Oxford Online Dictionary for refining the translation of this manuscript into English. These tools were utilized to enhance linguistic clarity and ensure precise terminology, guaranteeing both fluency and technical accuracy in the translated text.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sowmya, V.; Ragiphani, S. Air quality monitoring system based on artificial intelligence. In *Advances in Signal Processing and Communication Engineering: Select Proceedings of ICASPACE 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 267–273.
2. Zeng, J.; Liu, Y.; Ding, J.; Yuan, J.; Li, Y. Estimating On-Road Transportation Carbon Emissions from Open Data of Road Network and Origin-Destination Flow Data. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; Volume 38, pp. 22493–22501.
3. Szigetvári, T.; Túry, G. *Can They Get Out of the Middle-Income Technology Trap? State Strategies in Hungary and Türkiye in Promoting Automotive Investments*; IWE Working Papers No. 269; Institute for World Economics-Centre for Economic and Regional Studies: Budapest, Hungary, 2022.
4. Gábor, B. Assessing self-driving vehicle awareness in Hungarian rejecting groups. *Deturope* **2022**, *14*, 129–143. [CrossRef]
5. Guzmán-Torres, J.A.; Domínguez-Mota, F.J.; Tinoco-Guerrero, G.; García-Chiquito, M.C.; Tinoco-Ruíz, J.G. Efficacy Evaluation of You Only Learn One Representation (YOLOR) Algorithm in Detecting, Tracking, and Counting Vehicular Traffic in Real-World Scenarios, the Case of Morelia México: An Artificial Intelligence Approach. *AI* **2024**, *5*, 1594–1613. [CrossRef]
6. Rafique, S.; Gul, S.; Jan, K.; Khan, G.M. Optimized real-time parking management framework using deep learning. *Expert Syst. Appl.* **2023**, *220*, 119686. [CrossRef]
7. Almeida, A.; Fonseca, J.; Rasinhas, P.; Costa, C.; Paiva, D.; Silva, G.; Rito, P.; Sargento, S. Safe Roads: Traffic Management and Road Safety Platform for Smart Cities. In Proceedings of the 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), Aveiro, Portugal, 12–27 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–7.

8. Songire, S.; Patkar, U.; Shrivas, S.B.; Patil, U. Using YOLO V7 development of Complete VIDS solution based on latest requirements to provide highway traffic and incident real time info to the ATMS control room using Artificial intelligence. *SSRN* **2022**, 4313791. [CrossRef]
9. Ahmed, S.; Raza, M.; Kazmi, M.; Mehdi, S.; Rehman, I.; Qazi, S. Towards the next generation intelligent transportation system: A vehicle detection and counting framework for undisciplined traffic conditions. *Neural Network World* **2023**, *3*, 171. [CrossRef]
10. Yang, S.; Liu, Y.; Liu, Z.; Xu, C.; Du, X. Enhanced Vehicle Logo Detection Method Based on Self-Attention Mechanism for Electric Vehicle Application. *World Electr. Veh. J.* **2024**, *15*, 467. [CrossRef]
11. Tolba, M.A.; Kamal, H.A. SDC-Net++: End-to-End Crash Detection and Action Control for Self-Driving Car Deep-IoT-Based System. *Sensors* **2024**, *24*, 3805. [CrossRef]
12. Driessen, T.; Dodou, D.; Bazilinskyy, P.; De Winter, J. Putting ChatGPT vision (GPT-4V) to the test: Risk perception in traffic images. *R. Soc. Open Sci.* **2024**, *11*, 231676. [CrossRef]
13. Patil, O.; Nair, B.B.; Soni, R.; Thayyilravi, A.; Manoj, C. BoostedDim attention: A novel data-driven approach to improving LiDAR-based lane detection. *Ain Shams Eng. J.* **2024**, *15*, 102887. [CrossRef]
14. Rocha Filho, G.P.; Meneguette, R.I.; Neto, J.R.T.; Valejo, A.; Weigang, L.; Ueyama, J.; Pessin, G.; Villas, L.A. Enhancing intelligence in traffic management systems to aid in vehicle traffic congestion problems in smart cities. *Ad Hoc Netw.* **2020**, *107*, 102265. [CrossRef]
15. Musa, A.A.; Malami, S.I.; Alanazi, F.; Ounaies, W.; Alshammari, M.; Haruna, S.I. Sustainable traffic management for smart cities using internet-of-things-oriented intelligent transportation systems (ITS): Challenges and recommendations. *Sustainability* **2023**, *15*, 9859. [CrossRef]
16. Xiao, J.; Boschma, R. The emergence of artificial intelligence in European regions: The role of a local ICT base. *Ann. Reg. Sci.* **2023**, *71*, 747–773. [CrossRef]
17. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.H.; Lyu, S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]
18. Liang, L.; Ma, H.; Zhao, L.; Xie, X.; Hua, C.; Zhang, M.; Zhang, Y. Vehicle detection algorithms for autonomous driving: A review. *Sensors* **2024**, *24*, 3088. [CrossRef] [PubMed]
19. Weber, M.; Perona, P. Caltech Cars 1999 Dataset. 2002. Available online: https://data.caltech.edu/records/fmbpr-ezq86 (accessed on 17 May 2025).
20. Philip, B.; Updike, P.; Perona, P. Caltech Cars 2001 Dataset. 2001. Available online: https://data.caltech.edu/records/dvx6b-vsc46 (accessed on 17 May 2025).
21. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLOv8. Version 8.0.0, AGPL-3.0 License. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 16 February 2025).
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
23. Wang, J.; Zhang, W.; Cao, Y.; Chen, K.; Pang, J.; Gong, T.; Shi, J.; Loy, C.C.; Lin, D. Side-aware boundary localization for more precise object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part IV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 403–419.
24. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
25. Padilla, R.; Passos, W.L.; Dias, T.L.; Netto, S.L.; Da Silva, E.A. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
26. Sapkota, R.; Ahmed, D.; Churuvija, M.; Karkee, M. Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques. *IEEE Access* **2024**, *12*, 43436–43452. [CrossRef]
27. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
28. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
29. Wang, C.; Yang, W.; Zhang, T. Not every side is equal: Localization uncertainty estimation for semi-supervised 3D object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 3814–3824.
30. Liu, L.; Liu, X.; Gao, J.; Chen, W.; Han, J. Understanding the difficulty of training transformers. *arXiv* **2020**, arXiv:2004.08249.
31. Abibullaev, B.; Keutayeva, A.; Zollanvari, A. Deep learning in EEG-based BCIs: A comprehensive review of transformer models, advantages, challenges, and applications. *IEEE Access* **2023**, *11*, 127271–127301. [CrossRef]
32. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
33. Chen, X.; Yang, R.; Xue, Y.; Huang, M.; Ferrero, R.; Wang, Z. Deep transfer learning for bearing fault diagnosis: A systematic review since 2016. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–21. [CrossRef]

34. Pan, S.Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
35. Guo, H.; Zhang, Y.; Chen, L.; Khan, A.A. Research on vehicle detection based on improved YOLOv8 network. *arXiv* **2024**, arXiv:2501.00300. [CrossRef]
36. Zhou, J.; Xu, H.; Zhou, R.; Du, X. Based on Improved Lightweight YOLOv8 for Vehicle Detection. *Adv. Comput. Mater. Sci. Res.* **2024**, *1*, 293–300. [CrossRef]
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
38. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
39. Charef, A.; Jarir, Z.; Quafafou, M. Mobile Application Utilizing YOLOv8 for Real-Time Urban Traffic Data Collection. In Proceedings of the E3S Web of Conferences, London, UK, 20–22 August 2025; EDP Sciences: Les Ulis, France, 2025; Volume 601, p. 00077.
40. Bakirci, M. Real-time vehicle detection using YOLOv8-nano for intelligent transportation systems. *Trait. Signal* **2024**, *41*, 1727. [CrossRef]
41. Sun, H.; Fu, M.; Abdussalam, A.; Huang, Z.; Sun, S.; Wang, W. License plate detection and recognition based on the YOLO detector and CRNN-12. In *Signal and Information Processing, Networking and Computers: Proceedings of the 4th International Conference on Signal and Information Processing, Networking and Computers (ICSINC) 4th, Qingdao, China, 23–25 May 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 66–74.
42. Li, X.; Fu, C.; Li, X.; Wang, Z. Improved faster R-CNN for multi-scale object detection. *J. Comput.-Aided Des. Comput. Graph.* **2019**, *31*, 1095–1101. [CrossRef]
43. Jiang, D.; Li, G.; Tan, C.; Huang, L.; Sun, Y.; Kong, J. Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. *Future Gener. Comput. Syst.* **2021**, *123*, 94–104. [CrossRef]
44. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
45. de Andrade Porto, J.V.; Dorsa, A.C.; de Moraes Weber, V.A.; de Andrade Porto, K.R.; Pistori, H. Usage of few-shot learning and meta-learning in agriculture: A literature review. *Smart Agric. Technol.* **2023**, *5*, 100307. [CrossRef]