digital

MDPI

Article

SSAM_YOLOv5: YOLOv5 Enhancement for Real-Time Detection of Small Road Signs

Fatima Qanouni [1,*], Hakim El Massari [1,2,3], Noreddine Gherabi [1] and Maria El-Badaoui [1]

[1] National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga 25000, Morocco; h.elmassari@uca.ac.ma (H.E.M.); n.gherabi@usms.ma (N.G.); maria.el-badaoui@usms.ac.ma (M.E.-B.)
[2] Higher School of Technology, Cadi Ayyad University, El Kelâa des Sraghna 43000, Morocco
[3] LAMAI Laboratory, Faculty of Sciences and Techniques, Cadi Ayyad University, Marrakech 40000, Morocco
* Correspondence: qanouni.fatima@gmail.com

Abstract

Many traffic-sign detection systems are available to assist drivers with particular conditions such as small and distant signs, multiple signs on the road, objects similar to signs, and other challenging conditions. Real-time object detection is an indispensable aspect of these detection systems, with detection speed and efficiency being critical parameters. In terms of these parameters, to enhance performance in road-sign detection under diverse conditions, we proposed a comprehensive methodology, SSAM_YOLOv5, to handle feature extraction and small-road-sign detection performance. The method was based on a modified version of YOLOv5s. First, we introduced attention modules into the backbone to focus on the region of interest within video frames; secondly, we replaced the activation function with the SwishT_C activation function to enhance feature extraction and achieve a balance between inference, precision, and mean average precision (mAP@50) rates. Compared to the YOLOv5 baseline, the proposed improvements achieved remarkable increases of 1.4% and 1.9% in mAP@50 on the Tiny LISA and GTSDB datasets, respectively, confirming their effectiveness.

Keywords: traffic sign; small object; YOLOv5; real-time detection; attention module; SwishT_C

check for updates

Academic Editor: George A. Papakostas

Received: 29 May 2025
Revised: 15 July 2025
Accepted: 22 July 2025
Published: 29 July 2025

Citation: Qanouni, F.; El Massari, H.; Gherabi, N.; El-Badaoui, M. SSAM_YOLOv5: YOLOv5 Enhancement for Real-Time Detection of Small Road Signs. *Digital* **2025**, *5*, 30. https://doi.org/10.3390/digital5030030

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Road signs play a crucial role in traffic safety, giving key information to drivers and pedestrians. Recognizing and respecting these signs can significantly reduce incidences of accidents and road damage [1]. Autonomous vehicles assist drivers in detecting these signs through their use of advanced road-sign detection systems. Such systems must operate in real time; consequently, high speed and detection efficiency are two key criteria that influence the system's capabilities [2]. Speed refers to the capability of the model to examine and determine objects quickly, allowing adequate time for decision-making, while efficiency implies detection accuracy and, at the same time, an optimum use of computing resources, with the aim of achieving high precision with reduced latency.

Furthermore, the detection of road signs involves two major challenges that directly impact the quality of feature extraction. First, it is a small-object detection task, in which signs cover a limited number of pixels, contain minimal visual information [3], and are highly sensitive to disturbance. Second, the presence of objects similar to road signs can lead to ambiguities in interpretation [4], making detection more complex. To handle these

challenges, DL models have gained in importance in this field. DL extracts semantic information about objects and captures the contextual dependencies between them and the background [5]. YOLO is one of the approaches used to fix small-object detection issues in real time. YOLO is a single-stage detection network based on DL. Its first version was introduced in 2015 by Redmon et al. [6], and its basic principle is to divide an input frame into a series of regions and predict the bounding boxes and class probabilities for each region [7]. YOLOv5 was announced in 2020, in five sizes varying from nano to extra-large. It mainly uses the CSP backbone, which is designed to improve model speed, size, and accuracy by addressing the problem of duplicated gradient information and optimizing computation without losing significant information. FPN and PAN can also be applied [8]. The FPN [9] mechanism transmits strong semantic features from upper to lower feature maps. As for the PAN structure, it transmits strong localization features from inferior feature maps to superior feature maps [3].

The YOLOv5 network can detect road signs within a specific distance range. However, when a sign is very distant, the model fails to locate it. Our goal is to accurately detect various small traffic signs within a frame, regardless of their size variations. To address this challenge, we suggest an network which is improved in two main aspects: "where" to focus; and "how" to learn features. We evaluate the performance of our approach with different feature extraction enhancements using LISA and GTSDB datasets. By analyzing various feature extraction methods, this study focuses on finding the optimum balance between performance metrics for various application requirements.

We modified the activation function, and added an attention module as described below:

- First, we used SwishT_C, an improved version of the Swish activation function, by including a Tanh component to ensure consistent and reliable model learning, whatever the input values.
- Second, the SAM was integrated into the YOLOv5 backbone to optimize feature localization. Later, it was replaced by CBAM, which combines channel attention and spatial attention, to analyze performance of each attention module.
- Finally, we compared YOLOv5's performance with that of other proposed architectures, particularly in terms of precision, recall, mAP50, and inference speed.

The remaining sections of this paper are presented below, as follows: Section 2 reviews related work on small-object detection and feature extraction. Section 3 focuses on the architecture of the YOLOv5 network, specifically on the backbone of this network. In Section 4, we present the proposed method, highlighting the adapted activation function, the SAM module, and the CBAM module. Section 5 describes the experimental results and the impact of the proposed improvements. Finally, Section 6 concludes the article and describes our future research.

2. Literature Review

2.1. Small-Object Detection

Computer vision has been adopted in many real-world applications, such as intelligent surveillance systems, and self-driving cars. One of the most challenging tasks in the field is small-object detection. In fact, three methods of object detection are currently in use: two-stage detection, one-stage detection, and anchor-free algorithms [4].

Two-stage detection is divided into two parts [10]. In the first part, candidate regions are generated; in the second, these candidate parts are classified separately. Typical two-stage algorithms include Feature Pyramid Network FPN, Faster R-CNN, R-FCN, Cascade R-CNN, etc. In contrast, one-step detection algorithms such as YOLO and Single-Shot MultiBox Detector SSD [11] and their variants treat object detection as a regression problem by predicting bounding boxes and class probabilities of targets directly in a single step. The

YOLO algorithm improves detection efficiency by reducing false positives and improving both location accuracy and detection speed. The SSD algorithm gives the benefits of real-time detection and high accuracy. However, SSD is better adapted for the detection of large objects, and its performance is generally poor when applied to the detection of small objects [10]. Anchor-free algorithms avoid the use of anchor boxes and simply detect critical points. By presenting object detection as a keypoint detection task, this approach simplifies the overall process. However, it targets the most significant points without taking into account the object's internal features. Zhang et al. [12] improved the detection of industrial multiscale defects by employing anchor-free YOLOv5 with detailed sensitive PAN, with the aim of avoiding loss of local information.

In short, region-based methods are often highly accurate in terms of detection, but are not very applicable in real-time conditions. On the other hand, the one-stage methods developed to ensure a balance between detection speed and precision can satisfy real-time requirements [13].

2.2. Feature Extraction Methods

Feature extraction plays a central role in traffic-sign detection, particularly when dealing with small objects, diverse environmental conditions, and real-time performance constraints. Modern detectors such as YOLOv5 rely on a backbone network to extract semantic and spatial features at different resolutions. However, such detectors can suffer from low detection accuracy in challenging conditions. To address this, many recent studies have focused on feature extraction improvement. Some personalized object-detection systems have been designed to optimize feature extraction by focusing on edge information [14]. Liu et al. [15] suggested multiscale feature extraction using a pyramidal feature network based on bilateral attention that enhances feature learning through a combination of top-down and bottom-up sub-networks. Chu et al. [16] used self-attention to increase feature extraction and determine feature correlation. In addition, attention mechanisms such as CBAM attention [17–19] and light self-attention (LSA) modules [20] have been widely integrated to refine the extracted features by focusing on the most informative and interested regions. Another approach [21] allows a closer coupling of spatial and semantic information and an enhanced awareness of the global contextual scene through spatial and channel attention mechanisms.

Robust feature extraction can also address challenges related to detection speed and energy efficiency, both of which are essential in particular scenarios such as intelligent wireless surveillance. While deep learning models have achieved high accuracy in visual detection tasks, large model sizes and high computational requirements present significant challenges to their deployment on such systems. In their work on self-driving visual detectors, Choi et al. [22] presented a novel approach: a channel importance measure that combines detection saliency with special location information. Xue et al. [23] used a neural architecture search (NAS) method for object detection which was designed for YOLOv7 to reduce computational costs and search times by sharing and merging convolutional kernels. YOLOv9 is another alternative method for object detection. This improves feature extraction and reduces the loss of information inherent in deep neural networks through two key enhancements: a Generalized Efficient Layer Aggregation Network (GELAN) and Gradient Programmable Information (GPI) [24].

3. YOLOv5s Architecture

The YOLOv5s architecture consists of a backbone, a neck, and a head [25]. The backbone extracts features from the input frame at several levels of granularity using a convolutional neural network. The neck performs upsampling to obtain fine details, then

combines these features with those of the backbone. It then applies C3 blocks to enhance the extracted features. Finally, the head performs object detection at three different resolutions (small, medium, and large) as illustrated in Figure 1, generating bounding boxes and associated classes from the transmitted features.

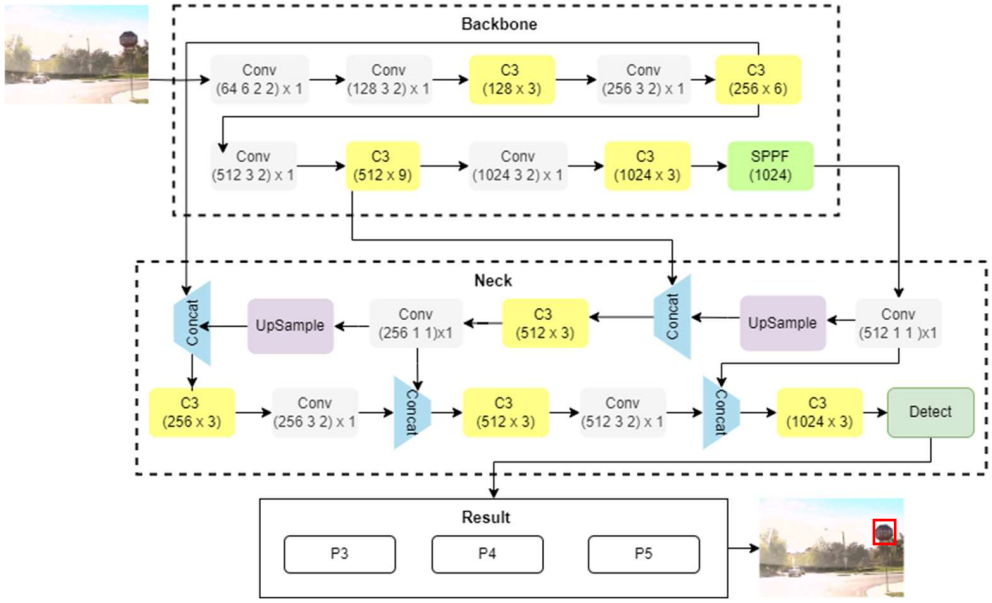

Figure 1. YOLOv5 architecture (The red box presents the output of YOLOv5).

The backbone network responsible for extracting features is designed on an enhanced Cross Stage Partial Daeknet53 CSPDarknet53 network that includes CSP connections to make learning more efficient and simplify model construction [26]. Additionally, attention modules are included in the C3 block, leading the model to focus on the most interesting regions of the input image. This enhancement improves detection accuracy by enabling the model to distinguish more precisely between important and less-important features.

The CSPDarknet53 architecture is a DenseNet-based model. It follows the CSPNet method, dividing the feature maps produced by the initial layer into two halves [27,28]. One of these halves passes through a dense block while the other is led directly to the next stage, as shown in Figure 2. This architecture delivers improvements in inference speed and detection accuracy. It is used to reduce redundant gradient information, computational costs, and model complexity while maintaining the good level of performance usually associated with DenseNet.

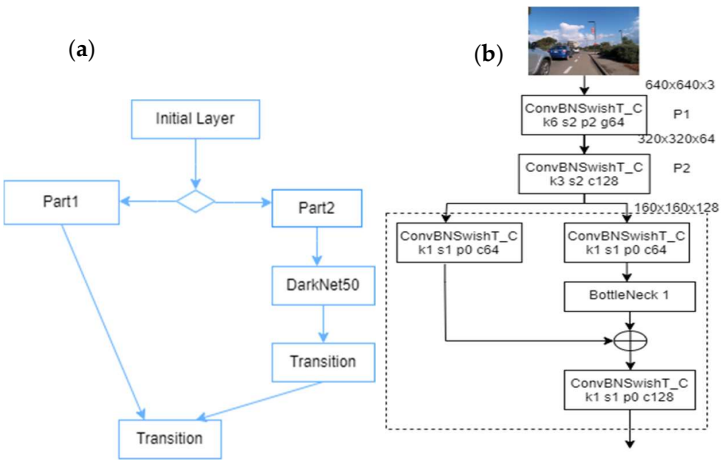

Figure 2. (**a**) CSPDarknet53 process. (**b**) The improved CSPDarknet53.

4. Method

To improve detection performance, we propose an enhanced version of YOLOv5, named SSAM_YOLOv5, which places more emphasis on improving feature extraction. The structure of our suggested network is illustrated in Figure 3.

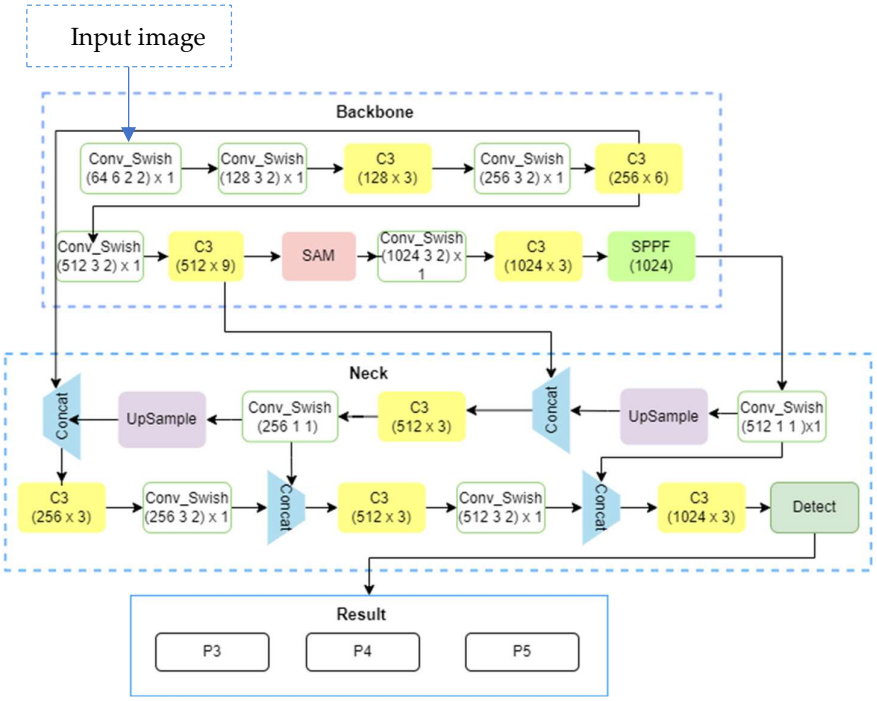

Figure 3. The improved YOLOv5 framework.

We applied two main changes to the YOLOv5 network. First, spatial information extraction was reinforced by adding an additional SAM, providing more detailed and meaningful features and often giving better detection performance. Second, we replaced the activation function, aiming to improve the model's learning capability.

4.1. Activation Function Enhancement

4.1.1. Swish T

Activation functions such as Sigmoid are essential for learning complex structures in data, thereby enabling smooth transitions. However, they are vulnerable to the problem of gradient vanishing. Swish introduces a dynamic choice between linear and non-linear modes, improving the ability of the neural network to adapt.

The Swish-T variants improve on the existing Swish activation function by combining it with hyperbolic tangent tanh bias, enabling greater use of negative values and producing a smoother, non-monotonic curve. This change gives a family of Swish-T Versions A, B, and C, each designed to perform at its best in different contexts. In this research we focused on SwishT_C.

Swish-T is based on adding α tanh(x) to the basic Swish function. This is expressed as follows:

$$\begin{aligned} F(x; \gamma; \alpha) &= x\sigma(\gamma x) + \alpha \tanh(x) \\ &= x\sigma(\gamma x) + \alpha(2\sigma(2x) - 1) \\ &= x\sigma(\gamma x) + 2\alpha\sigma(2x) - \alpha \end{aligned} \quad (1)$$

where $\sigma(x)$ indicates the sigmoid function, γ is a parameter that can be trainable, and α is a hyper parameter used to scale the tanh function from $(-1, 1)$ to $(-|\alpha|, |\alpha|)$.

For our improved YOLOv5, we used SwishT_C, the equation for which is as follows:

$$F_C(x; \gamma, \alpha) = \sigma(\gamma x)(x + 2\alpha\gamma) - \alpha/\gamma, \text{ where } \gamma = 0 \quad (2)$$

SwishT_C provides symmetry when γ changes sign, enabling stabilized performance across different input scales.

4.1.2. Gradient Expression of SwishT_C

The following Equation (3) illustrates the gradient expression of SwishT_C:

$$\frac{dFc}{dx} = \gamma\sigma\left(\gamma x\left(\left(1 - \sigma(\gamma x)\left(x + \frac{2\alpha}{\gamma}\right) + \sigma(\gamma x)\right)\right.\right. = \sigma(\gamma x)(\gamma(x - f(x)) + \alpha + 1) \quad (3)$$

The presence of $\alpha + 1$ in the last term indicates an improvement in gradient stability. This helps to avoid gradient vanishing and improve model performance. These advantages are essential to enhance learning performance in a complicated task such as small-road-sign detection.

4.2. SAM Module

The SAM focuses primarily on spatial information, i.e., the position of the target within the image. We applied average pooling (AvgPool) and maximum pooling (MaxPool) along the channel axis to the input feature map, in order to obtain two spatial feature maps Ms(F) of size $1 \times H \times W$. These two maps were then concatenated along the channel axis, forming maps of size $2 \times H \times W$. A convolution with a 7×7 kernel was then employed, followed by a sigmoid function to produce the final spatial attention map. This was then multiplied by the input feature map F to spatially weight important areas, as expressed in the following Equation (4):

$$M_S(F) = \sigma(f7 \times 7[\text{AvgPool}(F); \text{MaxPool}(F)]) \quad (4)$$

where σ is the sigmoid activation function, $f_{7 \times 7}$ is a convolution kernel with kernel size of 7×7.

4.3. CBAM Module

We replaced the SAM module with the CBAM module, helping the model to focus on the relevant regions in the input data and also enhancing the feature representation by removing irrelevant information. Convolution operations extract informative features by combining the Channel Attention Module CAM with the SAM so that each branch can learn "what" and "where" to focus on in the channel and spatial axes.

Considering an input feature map $F \in \mathbb{R}^{C \times H \times W}$, the module sequentially derives attention maps in two different dimensions, a 1D channel map $Mc \in \mathbb{R}^{C \times 1 \times 1}$ and a 2D spatial map $Ms \in \mathbb{R}^{1 \times H \times W}$, and then multiplies the attention maps to the input feature map to obtain more optimal features, as expressed in the following Equation (5):

$$F' = Mc(F) \otimes F,$$

$$F'' = Ms(F') \otimes F', \quad (5)$$

F'' is the final adjusted result. Figure 2 illustrates the calculation process for each of the attention maps.

5. Results and Discussion

The experiments were carried out using the following configuration: An Intel (R) Core (TM) i7-7600U CPU with a frequency of 2.90 GHz, and an NVIDIA Tesla T4 GPU with 16 GB of memory. The code was written and tested using the python language and the pytorch framework. When training the model, the input image size was set to 640 × 640, the model batch size was set to 32, and the number of training epochs per data iteration was set to 50.

In this section we introduce the evaluation of our proposed framework on two different datasets: Tiny LISA and GTSDB. Tiny LISA consists of 900 road-sign images divided into train, test, and validation sets, and classified into nine classes. GTSDB contains 900 traffic-sign frames divided into 600 training images and 300 evaluation images. These datasets are characterized by their diversity in terms of class, distance, and shape.

5.1. Ablation Experiments Comparison

We used the confusion matrix to evaluate and compute the performance of the improved model using precision, recall, and mAP@50 rates for each road-sign dataset. Metrics were calculated independently for each class [29]. Tables 1 and 2, along with Figures 4 and 5, summarize these metric results for the four models on both the Tiny LISA and GTSDB datasets.

Table 1. Ablation experiment results on the Tiny LISA.

Approach	Inference/ms	Recall	Precision	mAP@50
YOLOv5s (baseline)	-	88%	78%	86.7%
YOLOv5s + SAM	6.1	98.1%	73.5%	84.1%
YOLOv5 + SAM + SwishT_C (SSAM_YOLOv5)	9.1	98.1%	81.5%	88.1%
YOLOv5s + SwishT_C + CBAM	32	100%	78.3%	88.8%

Table 2. Ablation experiment results on GTSDB.

Approach	Inference/ms	Recall	Precision	mAP@50
YOLOv5s (baseline)	21.3	82.9%	89.6%	87.9%
YOLOv5s + SAM	24.6	81.8%	91.7%	86.8%
YOLOv5 + SAM + SwishT_C (SSAM_YOLOv5)	24.2	84.2%	92%	89.8%

The baseline YOLOv5 achieves recall, precision, and mAP@50 rates of 88%, 78%, and 86.7%, respectively, on the Tiny LISA dataset. By adding an additional SAM and incorporating the novel activation function, SSAM_YOLOv5 achieves improved values of 98.1%, 81.5%, and 88.1% for recall, precision, and mAP@50, representing increases of 10.1%, 3.5%, and 1.4%, respectively. Similarly, on the GTSDB dataset, SSAM_YOLOv5 achieves enhanced values of 84.20%, 92%, and 89.80% for recall, precision, and mAP@50, representing increases of 1.3%, 2.4%, and 1.90%, respectively, compared to YOLOv5.

Additionally, the CBAM with SwishT_C achieves a high mAP@50 rate of 88.8%, but the SSAM_YOLOv5 gives the best balance between recall, accuracy, and mAP@50, with a decent inference time of 9.1. This model is more stable, and involves less computational complexity than YOLOv5, YOLOv5 with SAM, or YOLOv5 with CBAM.

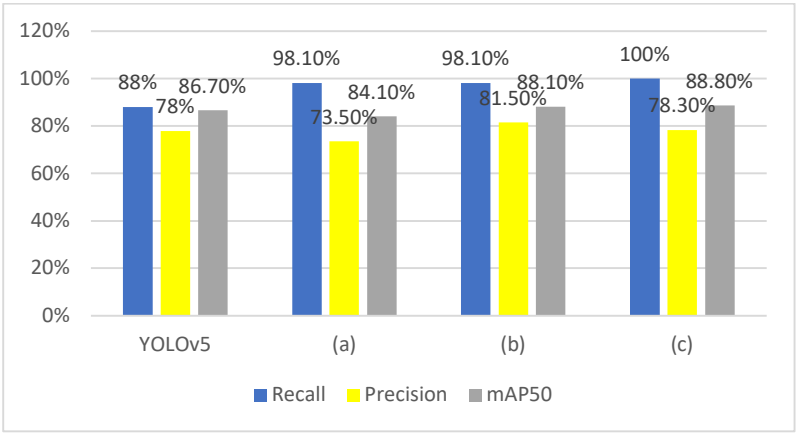

Figure 4. Recall, precision, and mAP@50 rates of our YOLOv5 variants using Tiny LISA. (**a**) YOLOv5s with SAM. (**b**) SSAM_YOLOv5. (**c**) YOLOv5s with SwishT_C and CBAM.

Graphical illustrations of precision, recall, mAP@50, and loss results for Tiny LISA are presented in Figures 6–8, in which it can clearly be observed that there is no overfitting or underfitting in the learning process. This shows that the deep learning process ran successfully.

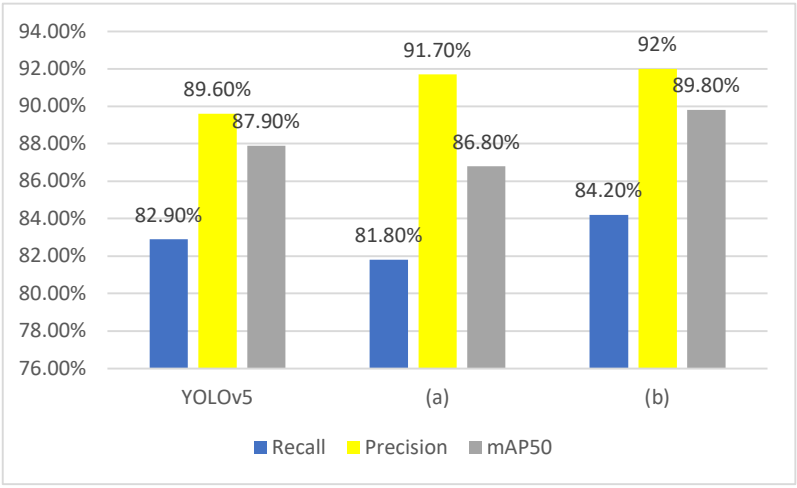

Figure 5. Recall, precision, and mAP@50 rates of our YOLOv5 variants using GTSDB. (**a**) YOLOv5s with SAM. (**b**) SSAM_YOLOv5.

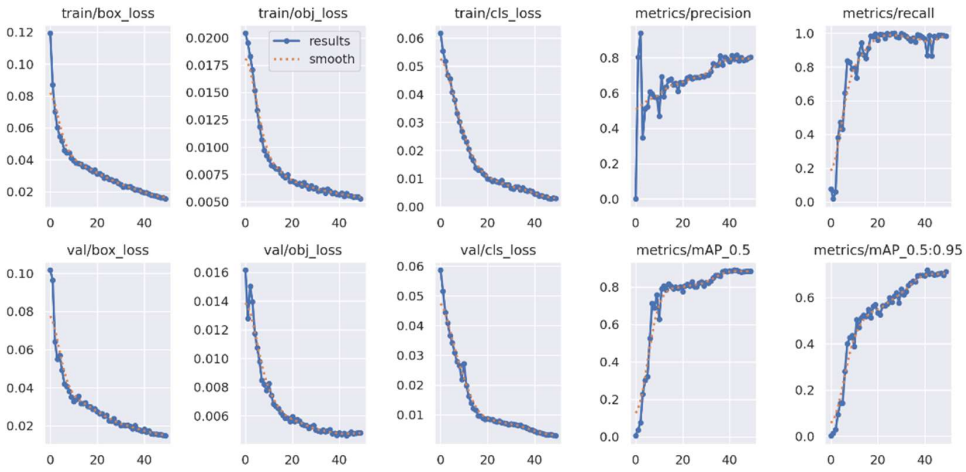

Figure 6. Evolution of Training and validation metrics of the baseline YOLOv5 on the Tiny LISA. The blue line shows how the model trains, and the orange line presents the smoothed values.

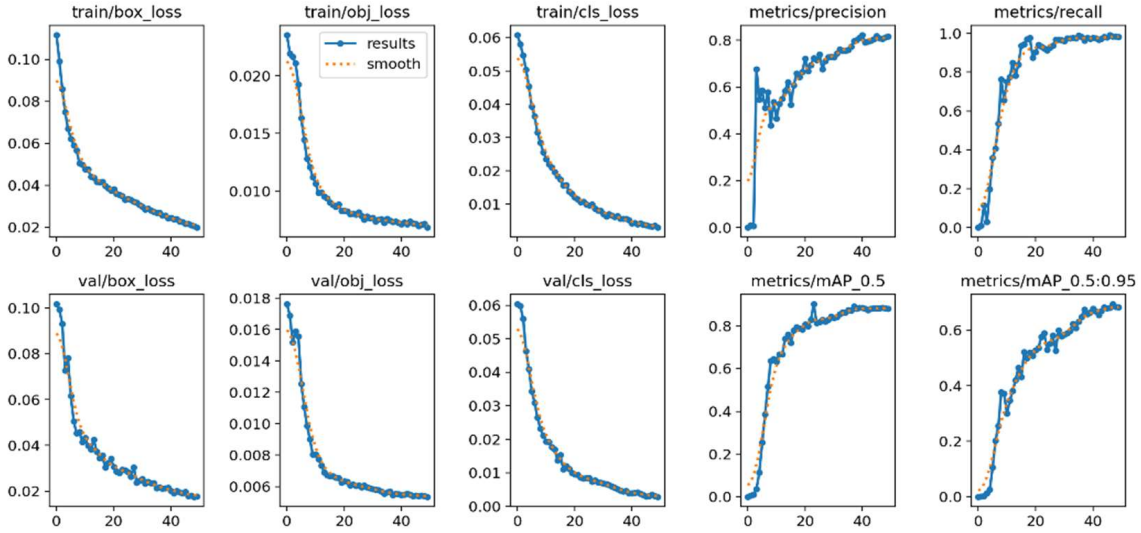

Figure 7. Evolution of Training and validation metrics of the SSAM_YOLOv5 on the Tiny LISA. The blue line shows the original results, and the orange line presents the smoothed values.

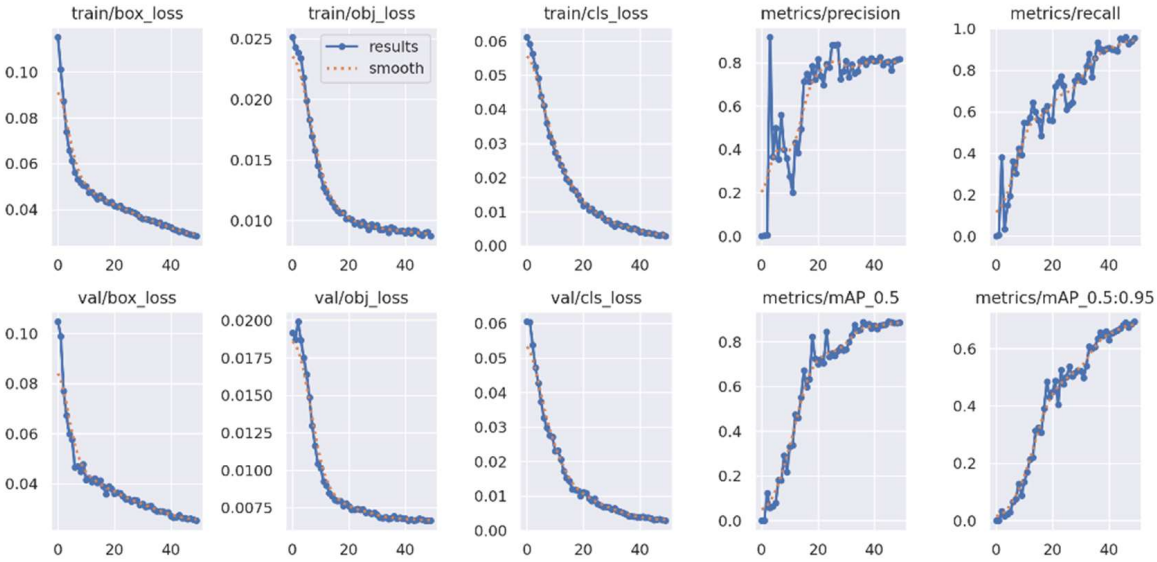

Figure 8. Evolution of Training and validation metrics of the CBAM-based YOLOv5 on the Tiny LISA. The blue line shows the original results, and the orange line presents the smoothed values.

The detection outcomes of baseline YOLOv5 and the improved YOLOv5 are illustrated in Figure 9, in which it can be seen that the applied improvements positively impacted the detection performance. Some distant road signs that are missed by baseline YOLOv5 are successfully detected by our models. Compared to the baseline YOLOv5, SSAM_YOLOv5 is better able to locate and accurately detect the distant traffic signs, demonstrating that the activation function SwishT_C enhances detection accuracy more effectively than SiLU.

Results of precision, recall, mAP@50, and loss of GTSDB are presented graphically in Figures 10–12, where the training curves are well converged. Moreover, the training and validation losses decrease exponentially over 50 epochs, indicating stable and effective learning. Additionally, the proposed approach can detect distant traffic signs that are not detected by baseline architecture, as shown in Figure 13.



Figure 9. Small-traffic-sign detection results on the Tiny LISA dataset using: (**a**) baseline YOLOv5, (**b**) SSAM_YOLOv5 module, (**c**) YOLOv5 with SAM, and (**d**) YOLOv5 with CBAM.

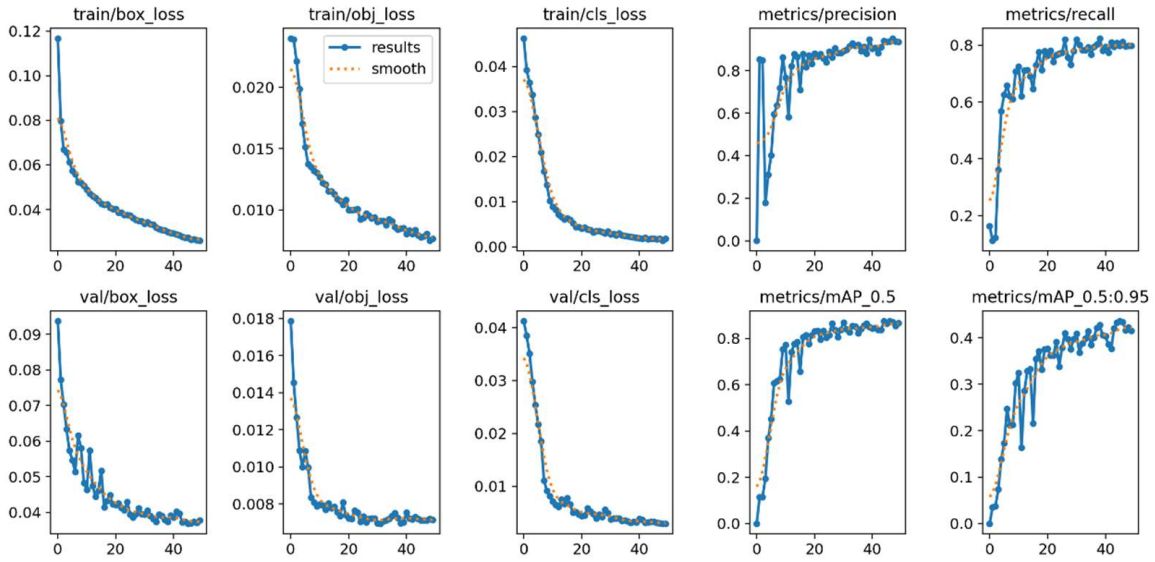

Figure 10. Evolution of Training and validation metrics of the baseline YOLOv5 on the GTSDB. The blue line shows the original results, and the orange line presents the averaged values.

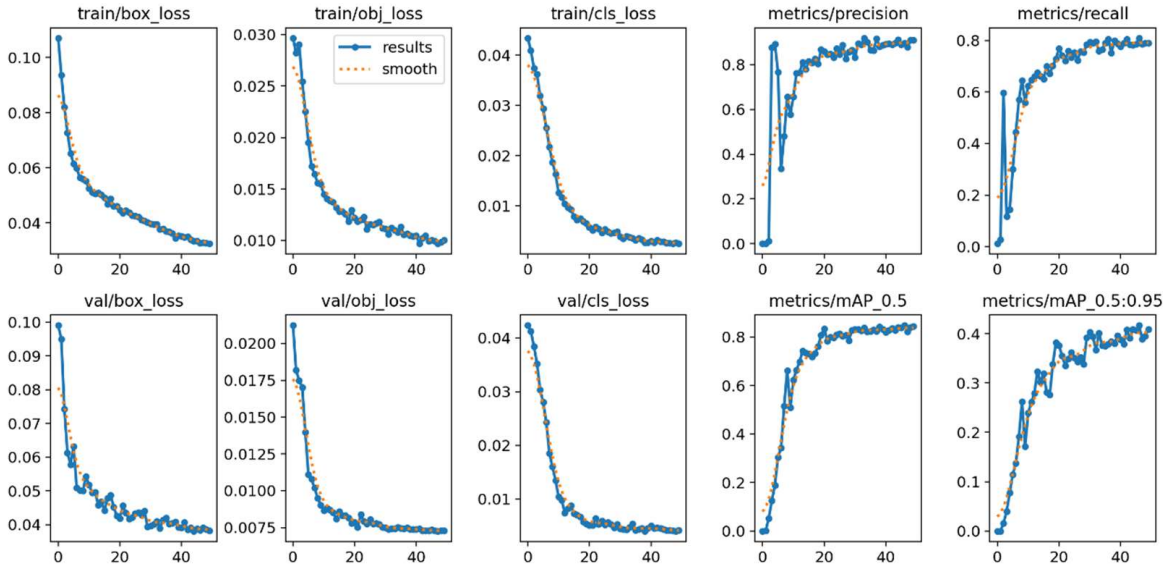

Figure 11. Evolution of Training and validation metrics of the SAM-based YOLOv5 on the GTSDB. The blue line shows the original results, and the orange line presents the averaged values.

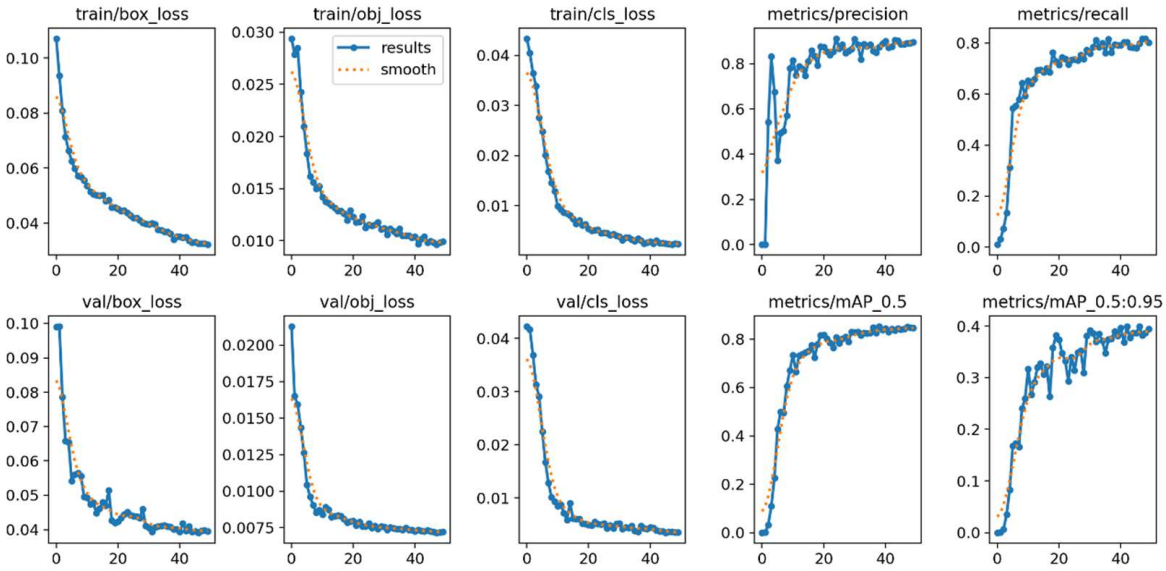

Figure 12. Evolution of Training and validation metrics of the SSAM_YOLOv5 on the GTSDB. The blue line shows the original results, and the orange line presents the averaged values.

5.2. Comparison of SSAM_YOLOv5 with Other Approaches

Table 3 shows a comparison of the detection performance of our approach with that of other methods. It can be seen that SSAM_YOLOv5 performs very competitively, achieving the highest mAP@50 (89.8%) while requiring far fewer training epochs (only 50) than the approach used by Zhang et al. [30]. Moreover, SSAM_YOLOv5 outperforms the model of Chu et al. [13] in terms of both accuracy and training efficiency, offering a higher mAP@50 with only a fraction of the training time. Although the model of [13] achieves a faster inference speed, SSAM_YOLOv5 provides a better balance between accuracy and computational cost, making it more appropriate for real-time traffic-sign detection where both accuracy and detection speed are critical.

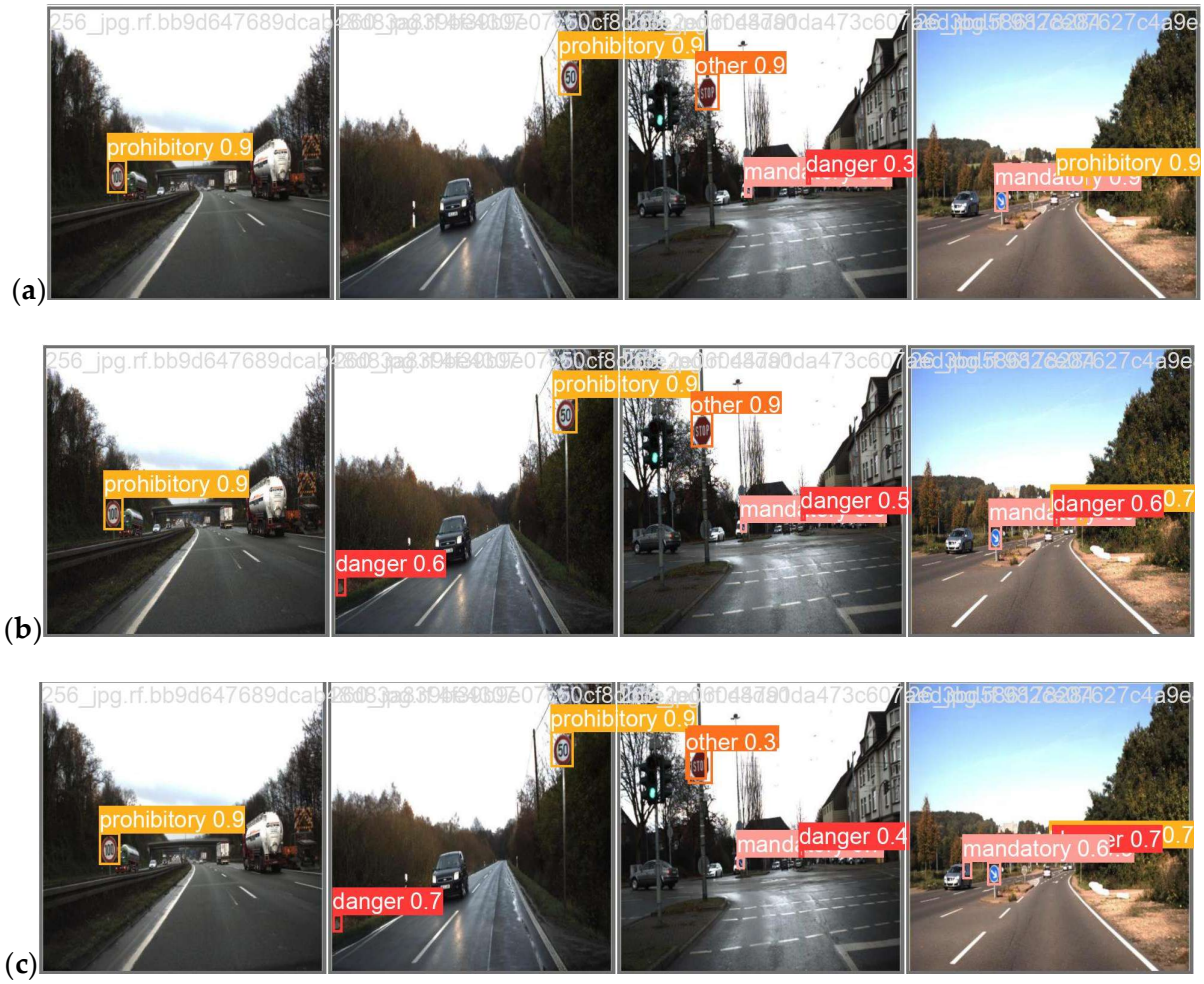

Figure 13. Small-traffic-sign detection results in GTSDB dataset using: (**a**) baseline YOLOv5, (**b**) YOLOv5 with SAM module, and (**c**) SSAM_YOLOv5.

Table 3. Comparison of the detection performance of our approach with that of different methods.

Approach	Image Size	Epochs	FPS	mAP@50
YOLOv5s (baseline)	640 × 640	50	46.95	87.9%
Wang et al. [19]	640 × 640	-	108	85.2%
Chu et al. [16]	640 × 640	600	73	86.5%
Zhao et al. [31]	640 × 640	120	37.45	88.2%
Zhang et al. [30]	640 × 640	300	78	90.1%
YOLOv5s + SAM	640 × 640	50	40.65	86.8%
SSAM_YOLOv5	640 × 640	50	41.32	89.8%

These results demonstrate that SSAM_YOLOv5 is a good candidate for real-time applications that require high precision with low inference times and training costs.

6. Conclusions

It is known that YOLOv5 is able to detect traffic signs in real time. However, it still struggles when detecting very distant signs. In this research, to address this challenge, we propose SSAM_YOLOv5. Our framework enhances the baseline YOLOv5 architecture with the aim of improving two important dimensions of real-time detection: detection speed and accuracy. To achieve this, we first replaced the SiLU activation function with a combination of Swish function and tanh bias to handle the learning process and increase detection

accuracy. Secondly, we integrated an additional attention module to focus more on small traffic signs. We compared the performance of YOLOv5 with that of three proposed variants. In the first variant, we added an SAM only. In the second, we replaced the SiLU activation function with SwishT_C. In the third, we replaced the SAM module with a CBAM module. The results showed that SwishT_C has a positive impact on the model's performance. Using CBAM, in terms of mAP@50, a value of 88.8% was achieved; however, 32 milliseconds were needed to detect a traffic sign. Otherwise, the SSAM_YOLOv5 proved to be the more stable model, giving the best balance between recall, accuracy, mAP@50, and inference speed on both the Tiny LISA and GTSDB datasets, along with reduced computational complexity.

Many environmental conditions can reduce the visibility of road signs. Bad weather conditions or poor lighting make road signs harder to detect. Additionally, the real-time detection of distant traffic signs across a sequence of frames can also decrease detection accuracy. These limitations underline the need for detectors that not only recognize target objects but also dynamically react to changing contexts. To overcome these challenges, detection systems must leverage robust feature-extraction methods that incorporate both spatio-temporal correlations and visual semantic features, using multi-source data such as video frames, sensor inputs, and environmental metadata. Our future work will involve improving context awareness through attention mechanisms and temporal correlation, in order to increase the generalization and the performance of the model under diverse and changing conditions.

Author Contributions: Conceptualization, F.Q. and M.E.-B.; methodology, F.Q.; software, F.Q.; validation, F.Q. and M.E.-B.; formal analysis, F.Q.; data curation, F.Q.; writing—original draft preparation, F.Q. and M.E.-B.; writing—review and editing, F.Q., N.G., and H.E.M.; visualization, F.Q. and H.E.M.; supervision, H.E.M. and N.G.; project administration, N.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the results of this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

CSP	Cross Stage Partial
PAN	Path Aggregation Network
FPN	Feature Pyramid Network
YOLOv5	You Only Look Once version 5
DL	Deep Learning
SSD	Single Shot Detector
SAM	Spatial Attention Module
CBAM	Convolutional Block Attention Module
SSAM_YOLOv5	SwishT_C with SAM in YOLOv5
LSA	Light Self-attention

References

1. Babić, D.; Babić, D.; Fiolic, M.; Ferko, M. Road Markings and Signs in Road Safety. *Encyclopedia* **2022**, *2*, 1738–1752. [CrossRef]
2. Seo, T.M.; Kang, D.J. PARA-CAM: Parallel Processing Architecture for Intelligent Real Time Multi IP Camera System with Deep Learning Models. *Int. J. Control Autom. Syst.* **2025**, *23*, 1563–1575. [CrossRef]

3. Horvat, M.; Jelečević, L.; Gledec, G. A Comparative Study of YOLOv5 Models Performance for Image Localization and Classification. In Proceedings of the Central European Conference on Information and Intelligent Systems, Faculty of Organization and Informatics, Varazdin, Croatia, 21–23 September 2022; pp. 349–356.
4. Peng, D.; Ding, W.; Zhen, T. A Novel Low Light Object Detection Method Based on the YOLOv5 Fusion Feature Enhancement. *Sci. Rep.* **2024**, *14*, 4486. [CrossRef] [PubMed]
5. Han, W.; Chen, J.; Wang, L.; Feng, R.; Li, F.; Wu, L.; Tian, T.; Yan, J. Methods for Small, Weak Object Detection in Optical High-Resolution Remote Sensing Images: A Survey of Advances and Challenges. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 8–34. [CrossRef]
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.
7. Jani, M.; Fayyad, J.; Al-Younes, Y.; Najjaran, H. Model Compression Methods for YOLOv5: A Review. *arXiv* **2023**. [CrossRef]
8. Khanam, R.; Hussain, M. What Is YOLOv5: A Deep Look into the Internal Features of the Popular Object Detector. *arXiv* **2024**. [CrossRef]
9. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
10. Du, L.; Zhang, R.; Wang, X. Overview of Two-Stage Object Detection Algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [CrossRef]
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016. [CrossRef]
12. Zhang, Y.; Zhang, H.; Huang, Q.; Han, Y.; Zhao, M. DsP-YOLO: An Anchor-Free Network with DsPAN for Small Object Detection of Multiscale Defects. *Expert Syst. Appl.* **2024**, *241*, 122669. [CrossRef]
13. Li, Z.; Dong, Y.; Shen, L.; Liu, Y.; Pei, Y.; Yang, H.; Zheng, L.; Ma, J. Development and Challenges of Object Detection: A Survey. *Neurocomputing* **2024**, *598*, 128102. [CrossRef]
14. Guiqiang, W.; Junbao, C.; Chengzhang, L.; Shuo, L. Edge-YOLO: Lightweight Multi-Scale Feature Extraction for Industrial Surface Inspection. *IEEE Access* **2025**, *13*, 48188–48201. [CrossRef]
15. Liu, Y.; Peng, J.; Xue, J.-H.; Chen, Y.; Fu, Z.-H. TSingNet: Scale-Aware and Context-Rich Feature Learning for Traffic Sign Detection and Recognition in the Wild. *Neurocomputing* **2021**, *447*, 10–22. [CrossRef]
16. Chu, J.; Zhang, C.; Yan, M.; Zhang, H.; Ge, T. TRD-YOLO: A Real-Time, High-Performance Small Traffic Sign Detection Algorithm. *Sensors* **2023**, *23*, 3871. [CrossRef]
17. Yan, J.; Zeng, Y.; Lin, J.; Pei, Z.; Fan, J.; Fang, C.; Cai, Y. Enhanced Object Detection in Pediatric Bronchoscopy Images Using YOLO-Based Algorithms with CBAM Attention Mechanism. *Heliyon* **2024**, *10*, e32678. [CrossRef]
18. Zeng, G.; Wu, Z.; Xu, L.; Liang, Y. Efficient Vision Transformer YOLOv5 for Accurate and Fast Traffic Sign Detection. *Electronics* **2024**, *13*, 880. [CrossRef]
19. Wang, Z.; Luo, W.; LI, X.; Hao, W. Traffic Sign Detection Based on Improved YOLOv5-S. In Proceedings of the 2024 China Automation Congress (CAC), Qingdao, China, 1–3 November 2024; pp. 740–745.
20. Chen, Z.; Yang, J.; Li, F.; Feng, Z.; Chen, L.; Jia, L.; Li, P. Foreign Object Detection Method for Railway Catenary Based on a Scarce Image Generation Model and Lightweight Perception Architecture. *IEEE Trans. Circuits Syst. Video Technol.* **2025**. [CrossRef]
21. Wang, X.; Guo, J.; Yi, J.; Song, Y.; Xu, J.; Yan, W.; Fu, X. Real-Time and Efficient Multi-Scale Traffic Sign Detection Method for Driverless Cars. *Sensors* **2022**, *22*, 6930. [CrossRef] [PubMed]
22. Choi, J.I.; Tian, Q. Saliency and Location Aware Pruning of Deep Visual Detectors for Autonomous Driving. *Neurocomputing* **2025**, *611*, 128656. [CrossRef]
23. Xue, Y.; Yao, C.; Wahib, M.; Gabbouj, M. YOLO-DKR: Differentiable Architecture Search Based on Kernel Reusing for Object Detection. *Inf. Sci.* **2025**, *713*, 122180. [CrossRef]
24. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**. [CrossRef]
25. Xu, Z.; Huang, X.; Huang, Y.; Sun, H.; Wan, F. A Real-Time Zanthoxylum Target Detection Method for an Intelligent Picking Robot under a Complex Background, Based on an Improved YOLOv5s Architecture. *Sensors* **2022**, *22*, 682. [CrossRef]
26. Abdimurotovich, K.A.; Cho, Y.-I. Optimized YOLOv5 Architecture for Superior Kidney Stone Detection in CT Scans. *Electronics* **2024**, *13*, 4418. [CrossRef]
27. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, New Orleans, LA, USA, 14–19 June 2020; pp. 390–391.
28. Senthil Kumar, K.; Abdullah Safwan, K.M.B. Accelerating Object Detection with YOLOv4 for Real-Time Applications. *arXiv* **2024**, arXiv:2410.16320.

29. Gledec, G.; Sokele, M.; Horvat, M.; Mikuc, M. Error Pattern Discovery in Spellchecking Using Multi-Class Confusion Matrix Analysis for the Croatian Language. *Computers* **2024**, *13*, 39. [CrossRef]
30. Zhang, H.; Liang, M.; Wang, Y. YOLO-BS: A Traffic Sign Detection Algorithm Based on YOLOv5. *Sci. Rep.* **2025**, *15*, 7558. [CrossRef]
31. Zhao, Q.; Guo, W. Small Object Detection of Imbalanced Traffic Sign Samples Based on Hierarchical Feature Fusion. *J. Electron. Imaging* **2023**, *32*, 023043. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.