sustainability

MDPI

Article

Enhancing Autonomous Driving Safety: A Robust Stacking Ensemble Model for Traffic Sign Detection and Recognition

Yichen Wang [1,†], Jie Wang [2,†] and Qianjin Wang [3,*]

[1] Division of Logistics and Transportation, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; wang-yc22@mails.tsinghua.edu.cn

[2] College of Eletronic and Information Engineering, Tongji University, Shanghai 2000092, China; 2054310@tongji.edu.cn

[3] School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China

* Correspondence: wyichen262@gmail.com

† These authors contributed equally to this work.

Abstract: Accurate detection and classification of traffic signs play a vital role in ensuring driver safety and supporting advancements in autonomous driving technology. This paper introduces a novel approach for traffic sign detection and recognition by integrating the Faster RCNN and YOLOX-Tiny models using a stacking ensemble technique. The innovative ensemble methodology creatively merges the strengths of both models, surpassing the limitations of individual algorithms and achieving superior performance in challenging real-world scenarios. The proposed model was evaluated on the CCTSDB dataset and the MTSD dataset, demonstrating competitive performance compared to traditional algorithms. All experiments were conducted using Python 3.8 on the same system equipped with an NVIDIA GTX 3060 12G graphics card. Our results show improved accuracy and efficiency in recognizing traffic signs in various real-world scenarios, including distant, close, complex, moderate, and simple settings, achieving a 4.78% increase in mean Average Precision (mAP) compared to Faster RCNN and improving Frames Per Second (FPS) by 8.1% and mAP by 6.18% compared to YOLOX-Tiny. Moreover, the proposed model exhibited notable precision in challenging scenarios such as ultra-long-distance detections, shadow occlusions, motion blur, and complex environments with diverse sign categories. These findings not only showcase the model's robustness but also serve as a cornerstone in propelling the evolution of autonomous driving technology and sustainable development of future transportation. The results presented in this paper could potentially be integrated into advanced driver-assistance systems and autonomous vehicles, offering a significant step forward in enhancing road safety and traffic management.

Keywords: autonomous driving; traffic sign detection and recognition; stacking ensemble model

Citation: Wang, Y.; Wang, J.; Wang, Q. Enhancing Autonomous Driving Safety: A Robust Stacking Ensemble Model for Traffic Sign Detection and Recognition. *Sustainability* **2024**, *16*, 8597. https://doi.org/10.3390/su16198597

Academic Editor: Elżbieta Macioszek

Received: 18 September 2024
Revised: 30 September 2024
Accepted: 1 October 2024
Published: 3 October 2024

1. Introduction

With the advancement of computer, communication, and intelligent signal processing technologies, Intelligent Transportation Systems (ITSs) are becoming increasingly popular. A cornerstone of ITSs is the intelligent detection of traffic signs, foundational to autonomous driving technology. Recent research underscores the importance of energy-efficient transportation infrastructures in smart cities, highlighting the need for innovative solutions in traffic and transportation engineering to reduce energy consumption and environmental impact [1]. Moreover, the integration of smart mobility concepts, such as roundabouts, offers new perspectives on safety and efficiency, particularly with the advent of connected and autonomous vehicles (CAVs) [2].

Traffic sign detection and recognition pose numerous challenges due to factors such as light intensity, weather conditions affecting images captured by vehicles on actual roads, and traffic sign targets typically occupying only a small portion of the entire image. From a technical perspective, firstly, applying traditional object detection technology to traffic sign detection and recognition proves challenging. Traditional object detection algorithms [3] employ exhaustive sliding window methods or image segmentation techniques to generate numerous candidate regions, subsequently extracting image features for each region, such as Histogram of Oriented Gradient [4] (HOG), Scale-Invariant Feature Transform [5] (SIFT), and Haar [6] features. These features are then fed into classifiers like Support Vector Machine [7] (SVM), Adaboost [8], and Random Forest [9] to categorize each candidate area. The computational overhead required to generate candidate regions, coupled with the limitations of features extracted by traditional methods, makes them ill-suited for traffic sign recognition, which demands high speed and accuracy in real-world applications. Thus, the accuracy and speed of traditional object detection methods fall short of autonomous driving requirements.

Secondly, traffic sign detection and recognition algorithms based on complex neural networks are also found to be inefficient. The introduction of AlexNet [10] by Krizhevsky et al. in 2012 significantly brought Convolutional Neural Networks (CNNs) into the spotlight. Following this, object detection algorithms based on deep neural networks have seen rapid development, offering higher accuracy at the expense of increased storage and computing power. However, the storage space and computing power available for traffic sign detection and recognition tasks are limited, as they are typically deployed on edge networks or mobile devices, making complex neural network-based algorithms unsuitable for this purpose.

Given the shortcomings of the aforementioned approaches, employing lightweight convolutional neural networks for traffic sign detection and recognition is currently viewed as the optimal strategy. Among single-stage object detection algorithms represented by You Only Look Once (YOLO) [11] and two-stage algorithms exemplified by Faster Region-based Convolutional Neural Network (Faster RCNN) [12], YOLO demonstrates superior accuracy and speed in the detection of close-range targets but lacks efficacy in detecting distant targets. In contrast, Faster RCNN achieves better precision in distant target detection but is less effective in global information extraction and recognition of proximate objects, with both stages being time-consuming and challenging for real-time detection. To surmount the limitations inherent in a singular algorithm for target detection, this paper integrates the Faster RCNN and YOLOX-Tiny models to devise a traffic sign detection and recognition algorithm for identifying traffic signs, aiming to enhance the speed and accuracy of driving safety warning feedback through model stacking ensemble. This paper makes contributions to the field in the following areas:

(1) We propose a traffic sign detection and recognition algorithm based on deep neural networks that performs object detection on road traffic signs and vehicles. Through the method of stacking ensemble, our deep neural network has excellently accomplished the task of traffic sign detection and recognition under multi-dataset testing, ensuring image processing efficiency while achieving ideal precision, and has demonstrated commendable robustness in quantitative analysis experiments and ablation experiments.

(2) By fusing the Faster RCNN and YOLOX-Tiny models, this structure is capable of establishing accurate and efficient traffic design detection. The efficacy of this method was assessed on the Changsha University of Science and Technology [13] (CCTSDB) dataset and Mapillary Traffic Sign Dataset [14] (MTSD) dataset, which encompass the vast majority of traffic signs found across China and globally. Data were categorized into five categories: close, distant, simple, moderate, and complex, and the model's performance in each category dataset was verified across multiple dimensions, including accuracy, recall rate, AP50, frame processing rate, and mAP. Results demonstrate the model's superior comprehensive performance over other benchmark methods, encompassing both single-stage and two-stage algorithms, including the weighted averaging ensemble YOLOX-Tiny-RCNN, YOLOX-Tiny, Faster RCNN, and SSD [15], notably leading in accuracy and complex scenarios against numerous algorithms.

(3) In the quantitative analysis experiments, we tested our model in scenarios including ultra-long-distance cases, multiple traffic signs in the same scene, partial occlusion cases, shadow interference cases, raindrop interference cases, and motion blur cases. The experimental results demonstrate that our model exhibits superior adaptability across various complex environments. Furthermore, in the ablation studies, we conducted an in-depth investigation of the model's convolutional layers and found that the model is also sensitively responsive to content in unrecognized signs that are similar to recognized content, further proving the robustness of our fusion model.

The rest of this paper is organized as follows. Section 2 discusses the current literature in this research area and identifies the shortcomings in existing solutions. Section 3 delves into the proposed stacking ensemble model and its technical architecture in detail. Section 4 discusses the comparative classification results of this deep neural network against other algorithms across multiple scenarios, including qualitative analyses and ablation studies. Finally, Section 5 concludes the paper.

2. Literature Review

In recent literature, numerous papers have been dedicated to the exploration of traffic sign detection and recognition. For this article, we conducted a search for journal articles on traffic sign recognition published between 2014 and 2024. Utilizing query strings such as traffic sign detection, traffic sign recognition, and traffic sign detection and recognition, over one hundred papers were identified. Each paper that provided detailed information and results on the recognition stages was read and analyzed in depth.

The works on traffic sign detection and recognition can be categorized into traditional methods and deep learning approaches. In traditional methodologies, features invariant to illumination, translation, scaling, and rotation are typically extracted using hand-crafted methods such as Histograms of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Bag of Words (BoW) [16], Local Binary Pattern (LBP) [17], and Haar. Subsequently, common classifiers employed at the classification stage include Support Vector Machines (SVM), Random Forest, or Artificial Neural Network (ANN) models, which are trained using these extracted features to classify the signs. The computational overhead required to generate candidate regions, coupled with the limitations of features extracted by traditional methods, makes them ill-suited for traffic sign recognition, which demands high speed and accuracy in real-world applications. On the contrary, deep learning approaches project the raw image into a feature space through the learning of a non-linear function, within which the categories are linearly separable. However, deep learning methods necessitate extensive computational resources and millions of labeled data to achieve satisfactory results.

In recent years, numerous researchers have explored the use of CNN-based deep learning methods to address the Traffic Sign Detection and Recognition problem, such as TSD-YOLO [18] (Zhao et al., 2024), Faster RCNN [12] (Ren et al., 2017), Mask RCNN [19] (Megalingam et al., 2023), CNN [20] (Rani et al., 2024), and YOLOv7 [21] (Ren et al., 2024) models. However, the RCNN model consists of different blocks for classification and regression, which augments the computational demand. SSD- and YOLO-based models are less computationally intensive but demonstrate limited accuracy in the detection of small traffic signs.

In the advancement of classic traffic sign detection algorithms, numerous scientists both domestically and internationally have employed diverse technologies in integration, aiming to achieve enhanced accuracy. L. Chen et al. (2017) [19] introduced a composite convolutional neural network, comprising two independent CNNs: one designated for superclass recognition encompassing six categories, and the other for subclass recognition covering 43 categories. The final label is determined through a vector summation of the outcomes from both CNNs. The framework proposed in their study achieved a recognition accuracy of 95.6%. However, the temporal cost attributed to the proposed framework was 2.7 milliseconds. Despite the results being acceptable, the classification time is considered too slow for real-time applications. Kedkarn et al. (2015) [22] also utilized techniques like SVM to classify traffic signs from the German Traffic Sign Recognition Benchmark (GTSRB). Kumar et al. (2020) [23] proposed an improved SSD (single shot multibox detector) algorithm based on a multi-layer convolutional neural network. By selecting the optimal aspect ratio of the convolution kernel, the precision of target detection has been effectively improved. However, due to the algorithm's reliance on the default box's adaptability to the scene, its real-time performance is difficult to guarantee. Tabernik et al. (2020) [24] improved the Mask RCNN algorithm for large-scale road sign detection and recognition issues, proposing a distributed data augmentation technique based on geometric appearance and visual distortion, but the precision of the algorithm decreased due to loss of data in the classification network. Li et al. (2021) [25] presented a cross-layer fusion method for multi-object detection and recognition based on an improved Faster RCNN model, suitable for complex traffic environments. This method uses the five-layer structure of VGG16 to obtain more feature information, achieving higher precision in small object target detection, but it did not address the issue of Faster RCNN's poor global information extraction and close object recognition performance. Carrasco et al. (2023) [26] introduced a method for targeting tiny road objects from a holographic perspective, adjusting the YOLO-v5 model with a multi-scale module and spatial attention mechanism, thus improving the YOLO model's precision in recognizing small objects, but this algorithm only significantly improves precision in specific scenario detections and lacks robustness.

Currently, related research often completes target detection through single-category models. Although improvements to the models have enhanced detection effects to some extent, it is difficult to balance detection precision and efficiency in actual road condition monitoring. Among the single-stage target detection algorithms represented by YOLO and the two-stage target detection algorithms represented by Faster RCNN, YOLO performs better in terms of precision and speed for close-range target detection but is less effective for detecting distant objects, whereas Faster RCNN has better precision for distant target detection but is weaker in global information extraction and close object recognition, and the two-stage algorithms are time-consuming, making real-time detection difficult to achieve.

From recent research, we find that traditional methods are inadequate for real-time traffic sign recognition due to their computational constraints. Deep learning models, especially CNN-based ones, have shown improved performance but are hindered by the need for significant computational power and large annotated datasets. The trend towards hybrid models suggests a potential path forward in enhancing detection accuracy and computational efficiency. To overcome the limitations of single-algorithm target detection, this paper integrates the Faster RCNN and YOLOX-Tiny models to design a traffic sign detection and recognition algorithm for target detection of traffic signs and vehicles with the hope of improving the speed and accuracy of driving safety warning feedback through model ensemble.

3. Methodology

In this study, we propose a traffic sign detection and recognition model based on the Stacking Ensemble (SE) learning method, aimed at addressing the main shortcomings of current solutions, namely the difficulty in simultaneously ensuring recognition accuracy and efficiency. We opted for stacking (Rashid et al., 2020) [27] over other popular ensemble models such as bagging (Kotsiantis et al., 2007) [28] and boosting (Webb and Zheng, 2004) [29] because the latter two methods are often used to create homogeneous ensembles, which is an ensemble formed from the same type of classifiers (Aburomman and Reaz, 2016) [30]. Additionally, SE is a machine learning technique that can learn how to combine heterogeneous weak base-learners using a meta-learner, thus achieving optimal recognition accuracy. It forms a hierarchy of different machine learning models by utilizing the predictions of different classifiers from the previous layer, akin to the deep learning approach. This ensures the diversity of the base classifiers, thereby leading to enhanced model performance (Aburomman and Reaz, 2016; Tama and Rhee, 2017; Zhou et al., 2020) [30–32].

Base-learners and meta-learners constitute the level-0 and level-1 classifiers in an SE model, respectively. Compared to single machine learning models, this method demonstrates improved performance and increased robustness (Zhou et al., 2020; Sarmas et al., 2022; Zhang et al., 2021 [32–34]); different machine learning models prioritize various input features based on their operational mechanisms to make class predictions.

3.1. Faster RCNN Model

The Faster RCNN model is one of the most effective general-purpose target detection algorithms in the RCNN series. During driving, it is crucial for drivers to be aware of road conditions ahead. However, most target detection algorithms have low efficiency in detecting small objects, sometimes failing to recognize them. By employing the Faster RCNN network structure, not only is it possible to detect small objects where previously there were none, but there is also a significant improvement in the accuracy of small object detection, with a tolerance rate within a reasonable range. The Faster RCNN network structure includes four modules: feature extraction network, Region Proposal Network (RPN) [12], Region of Interest (ROI) pooling, and classification and regression. Images of fixed size M×N are fed into the feature extraction network, utilizing 13 convolutional (conv), 13 Rectified Linear Unit (ReLU), and 4 pooling layers to extract feature maps. These feature maps are then input into the RPN, generating multiple regions of interest after 33 convolutions. The RPN structure classifies anchor points as positive or negative using the Softmax function, then calculates the bounding box regression offsets for precise candidate selection. The candidate layer integrates positive anchors and corresponding bounding box regression offsets to obtain adjusted candidates, eliminating those too small or beyond boundaries. Adjusted candidates are projected onto the feature map and scaled to 7 × 7 region feature maps through the ROI pooling layer, then flattened into fully connected layers. Class probabilities are computed via the softmax function, and object positions are precisely adjusted through bounding box regression, finally outputting the object's category and precise location in the image.

3.2. YOLOX-Tiny Model

YOLOX-Tiny, a lightweight version of YOLOX, features a simplified structure and faster detection speed. Its algorithmic framework is divided into the backbone, neck, and detection head. Compared to the standard model, it weakens Mosaic and removes Mix, yielding good detection performance even on less powerful hardware platforms, thus improving algorithm compatibility [35]. The model consists of a convolutional space module (CSM), convolutional space unit (CSU), and convolutional component unit (CCUCR). CSM, composed of convolution layers, Sigmoid functions, and Mul functions, extracts low-level image features like boundaries and colors. CSU, the feature extraction layer, includes two CSM layers and an Add function, extracting high-level image features such as shapes and structures. CCUCR, the classification layer, identifies different parts of the target to determine its position and category. In the YOLOX-Tiny network structure, images sized M×N are processed through slicing operations and fed into the feature extraction network. After extracting image features through two CSM layers, classification is performed using four CSM layers, two CCUCR layers, and an additional two CSM and four CCUCR layers. All CCUCR layers are followed by upsampling to ensure detail features and resolution, thereby achieving better detection of small objects. Finally, features are fused and transposed for output.

3.3. Model Ensemble

3.3.1. Stacking Ensemble

In this study, we employ two base-learners, namely (1) YOLOX-Tiny, and (2) Faster RCNN. Single-stage object detection algorithms like YOLOX-Tiny can meet the requirements for accuracy and speed in close-range target detection tasks but perform poorly in detecting distant targets. On the other hand, Faster RCNN offers better precision for distant targets but is weaker in global information extraction and close-range recognition; in addition, the two-stage algorithms are time-consuming, making it difficult to ensure sufficient response time in real-time autonomous driving recognition tasks.

A meta-learner is usually utilized to intelligently combine multiple predictions to reduce incorrect detection by employing a single classifier. In our proposed model, the logistic regression classifier is chosen as the meta-learner due to its simplicity and interpretability, optimally integrating the diverse characteristics of the machine learning models to reduce false alarms and enhance detection accuracy and efficiency. In summary, the Stacking Ensemble (SE) model proposed in this article is implemented using the Scikit-Learn library (Pedregosa et al., 2011) [36] within the Python programming language. The machine learning models utilized as base learners include Faster RCNN (faster regions with CNN features) and YOLOX-Tiny (You Only Look Once X-Tiny). Furthermore, Logistic Regression is employed as the meta-learner, which takes the predictions from the base learners as input and then makes the final prediction [37].

As is shown in Figure 1, during the training phase, a 5-fold cross-validation approach is adopted, allowing the base learners to be trained on 4 folds of the training data, while making predictions on the 5th fold. In the first layer of training, we use traditional Faster RCNN and YOLOX-Tiny as base learners to produce the results of the first layer training, which will then be used for secondary training by the logistic regression in the second layer. Layer 2 uses the prediction results of the base learner in Layer 1 as training data for a new round of predictions with a logistic regression algorithm. This process is iterated to achieve predictions corresponding to the entire training set; the specific iteration process is illustrated in Figure 2. Firstly, the traffic sign dataset is collected and preprocessed, including image scaling, normalization, and data augmentation, followed by classifying the data to adapt to model training. At the same time, considering the impact of external factors such as lighting, weather, and occlusions on traffic sign detection, relevant data are collected to assess the robustness of the model. The dataset is innovatively divided into categories such as distant, close, complex, and simple for training, thereby simulating different road conditions. During the model-training phase, the accuracy and efficiency of the model are improved by adjusting the network structure and optimizing training strategies, and the model is verified and tested on different datasets to ensure its effectiveness in practical applications.

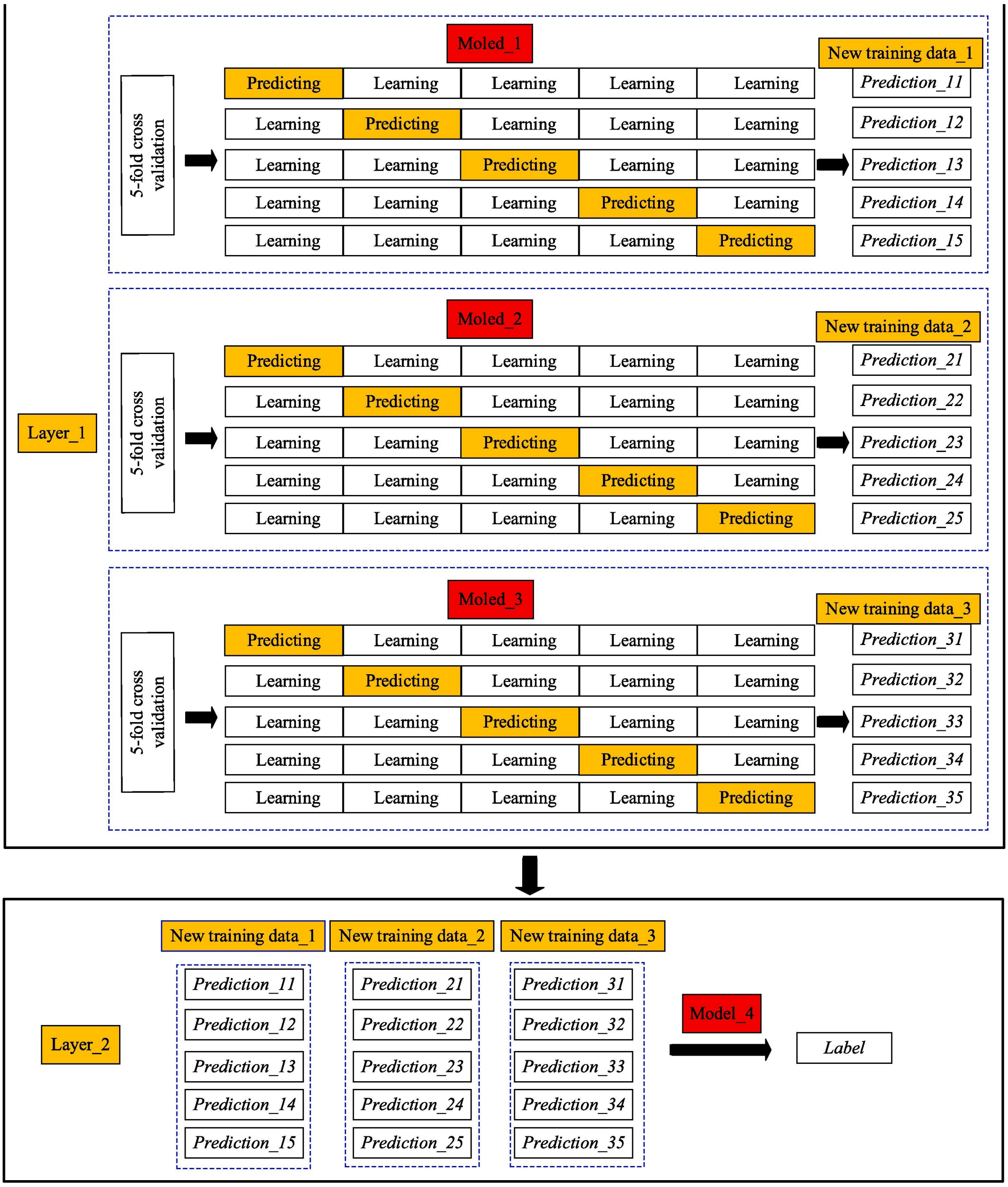


Figure 1. Schematic representation of the stacking ensemble model.

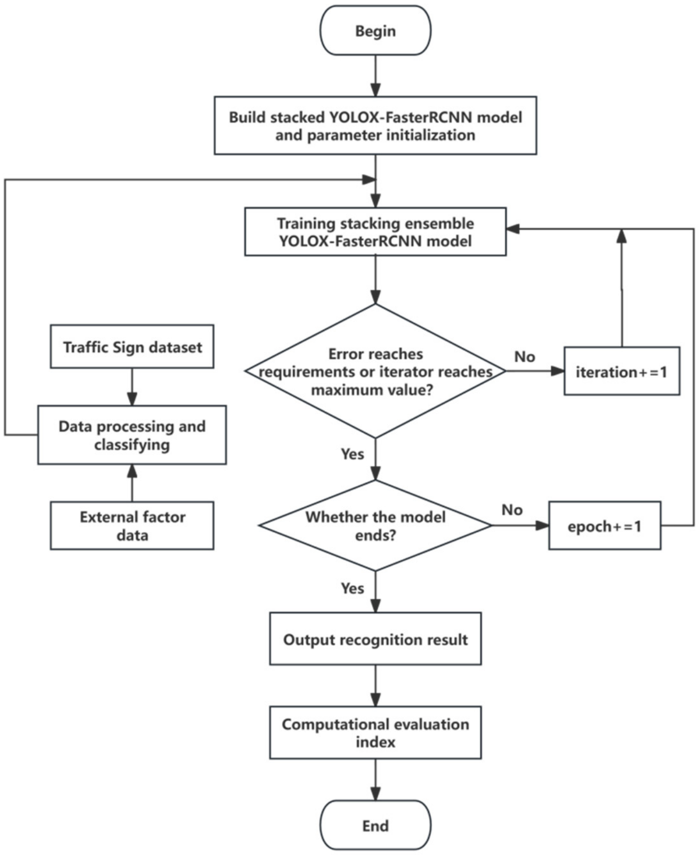

Begin
Build stacked YOLOX-FasterRCNN model and parameter initialization
Training stacking ensemble YOLOX-FasterRCNN model
Traffic Sign dataset
Error reaches requirements or iterator reaches maximum value?
No
iteration+=1
Data processing and classifying
Yes
External factor data
Whether the model ends?
No
epoch+=1
Yes
Output recognition result
Computational evaluation index
End

Figure 2. Flowchart of the iterative structure of the ensemble model.

3.3.2. Weighted Averaging Ensemble

Due to the fundamental differences between these two categories of algorithms, the resulting accuracy and speed of outcomes vary. To ensure the accuracy and completeness of this paper, we have also designed a weighted averaging model ensemble method as a control experiment against the stacking ensemble method, aiming to identify the optimal model fusion approach. The formula for the weighted average fusion method is as follows: $H(x) = \sum_{i=1}^{T} w_i \hbar_i(x)$, where H denotes the prediction result of the fused model, x represents the input image, T is the number of individual learners, w_i is the weight of the i_{th} individual learner, and $\hbar_i$ is the prediction result of the i_{th} individual learner.

Figure 3 illustrates the specific structure of model fusion. Images with pixel dimensions P×Q are scaled to M×N, followed by detection through YOLOX and Faster RCNN models individually. The detection results from each model are then computed with preset weights to derive the final outcome. YOLOX-Tiny demonstrates high detection accuracy and speed for close-range targets but may not detect distant targets effectively. Therefore, in this model, distant target detection primarily relies on the Faster RCNN model, while close-range recognition depends on the YOLOX-Tiny model. The weights of the two models were tested and determined using the CCTSDB dataset. When the target is at a moderate distance, the fusion model’s detection accuracy, measured by Recall, mean Average Precision (mAP) and Frames Per Second (FPS), was tested with different weights for the YOLOX-Tiny model ranging from 0.6 to 0.9. Table 1 shows that when the YOLOX-Tiny weight is set at 0.75, the fusion model performs best.

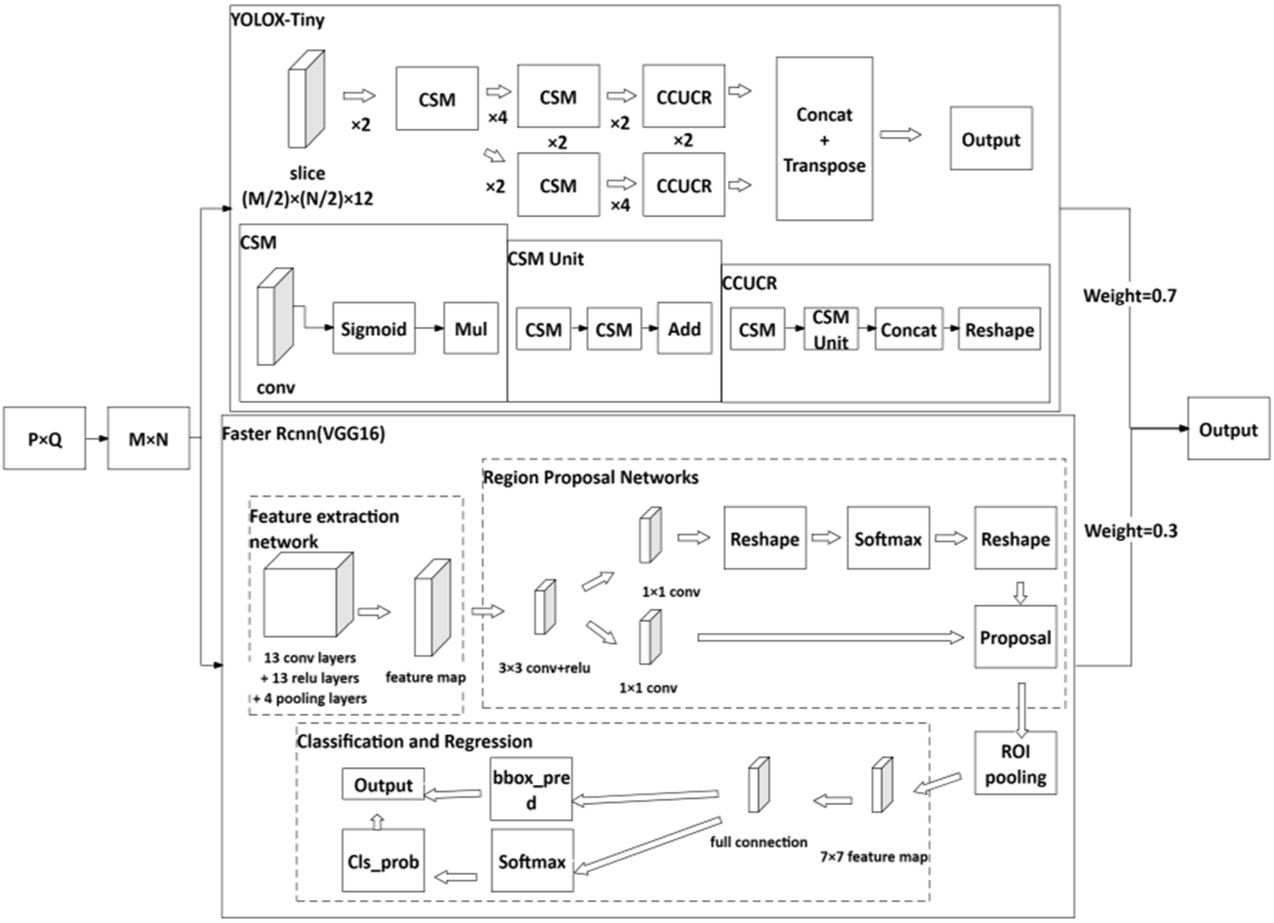


Figure 3. Schematic diagram of the weighted averaging ensemble model structure.

Table 1. Performance of the ensemble model with different weights for YOLOX-Tiny on the CCTSDB dataset.

Weight	Recall	MAP	FPS
0.60	85.3%	84.4%	85.0
0.65	86.8%	85.8%	85.3
0.70	87.4%	86.5%	86.9
0.75	92.9%	90.2%	87.1
0.80	88.4%	87.3%	83.3
0.85	87.9%	85.5%	84.3

3.4. Evaluation Standard

To evaluate the performance of the model in detecting five different categories of road traffic targets, Precision (P), Recall (R), mean Average Precision (mAP), and Frames Per Second (FPS) were employed for quantitative analysis. The calculations for precision and recall are presented in Equations (1) and (2), respectively. Herein, a True Positive (*TP*) is a positive sample predicted as positive by the model, a False Positive (*FP*) is a negative sample predicted as positive by the model, a True Negative (*TN*) is a negative sample predicted as negative by the model, and a False Negative (*FN*) is a positive sample predicted as negative by the model.

The mean Average Precision is calculated based on precision and recall values. In tasks involving multi-type target detection, the detection precision of the model is assessed by calculating the mAP across all types, the value of mean Average Precision equals the area under the Precision-Recall (P-R) curve, with higher values indicating greater accuracy of the network. This calculation is detailed in Equation (3).

In addition to detection accuracy, operational speed is another crucial performance metric for target detection algorithms. A common measure of speed is FPS, which denotes the number of images processed per second, as expressed in Equation (4). In our study, FPS was evaluated using a single NVIDIA GeForce 1080Ti graphics card.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{mean Average Precision} = \frac{1}{\text{classes}} \sum_{\text{i=1}}^{\text{classes}} \int_{0}^{1} \text{P(R)dR} \tag{3}$$

$$\text{Frames per Second} = \frac{N}{\sum_{j}^{N} T_j} \tag{4}$$

4. Experiments

In our experiments, we conducted comparative analyses by juxtaposing our fusion model against weighted averaging ensemble YOLOX-Tiny-RCNN, YOLOX-Tiny, Faster RCNN, SSD, and YOLO v3-Tiny algorithms. These evaluations were performed on the CCTSDB dataset, which is segmented into five categories based on the proximity and complexity of the subjects, to objectively assess the merits and demerits of our model. Beyond leveraging our methodology on the CCTSDB dataset, which is a resource extensively utilized within the Chinese scholarly community, for cross-dataset experimentation, we also employed the MTSD dataset for training purposes due to its inclusion of traffic signs from a diverse array of countries. All experiments were conducted using Python 3.8 on the same system equipped with an NVIDIA GTX 3060 12G graphics card. Lastly, we evaluated the performance of our model in scenarios including ultra-long-distance cases, multiple traffic signs in the same scene, partial occlusion cases, shadow interference cases, raindrop interference cases, and motion blur cases. The detailed outcomes and their respective discussions are delineated in the subsequent sections.

4.1. Results on CCTSDB

This study employed the Changsha dataset for model training, which includes five categories: indication signs, prohibition signs, warning signs, non-motor vehicles, and motor vehicles. The CCTSDB 2021 dataset comprises 20,492 images, divided into a training set and a positive sample test set, with traffic signs in the images categorized by their meaning into indication signs, prohibition signs, warning signs, non-motor vehicles, and motor vehicles. The training set contains 18,992 images, numbered from 00,000 to 18,991, while the positive sample test set contains 1500 images, numbered from 18,992 to 20,491. The negative sample includes 500 negative sample images.

The dataset was divided into training and testing sets at a ratio of 8:2; the specific allocation of sign quantities is illustrated in Figure 4. The training platform used was Python 3.8, and the graphics card was an NVIDIA GTX 3060 12G.

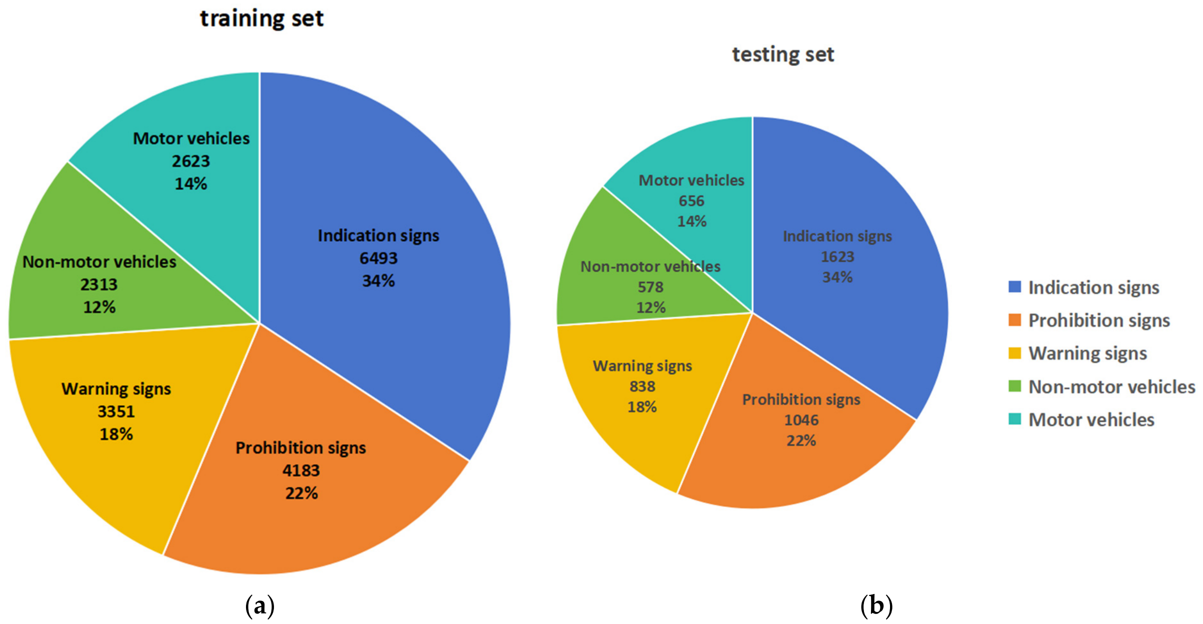


Figure 4. Number of traffic signs per category for (**a**) training and (**b**) testing data in CCTSDB dataset.

To evaluate the detection capabilities of the algorithm, a comparative experiment was conducted between the model proposed in this paper and the weighted averaging ensemble YOLOX-Tiny-RCNN, YOLOX-Tiny, Faster RCNN, SSD, YOLO v3-Tiny, Improved YOLOv4-Tiny [38], and C2Net-YOLOv5 [32] algorithms. Each model underwent 200 epochs of training, followed by predictions on data pertaining to distant and close targets, as well as simple, moderate, and complex scenes. Distant targets were defined as occupying the upper third of the camera-captured image, while close targets occupied the lower two-thirds; a highway with vehicles only ahead constituted a simple scene, a highway with vehicles both ahead and to the side a moderate scene, and a highway with vehicles ahead, to the side, and with traffic signs a complex scene. We empirically set a learning rate of 0.00261, which is increased after training for 90% iterations. The input image size is set to 640 × 640 pixels to reduce the training time. For better feature extraction, image resolution is increased during testing. Hence, final results are reported on the image with resolution 1024 × 1024.

Table 2 provides a detailed comparison of the performance of several different traffic sign detection models under various scenarios. Our paper model achieves high average precision (AP50%) across all scenarios, reaching 86.1% in complex scenarios, and leads with a score of 88.5% in mean average precision (MAP), indicating that this model has excellent generalization and robustness in handling various traffic sign detection tasks. Additionally, its FPS reaches 90.7, demonstrating outstanding real-time processing capabilities. The weighted averaging ensemble model performs slightly less well than the paper model, but still maintains a high AP50% across all scenarios, especially in distant and simple scenarios where its performance is close to the paper model. This may be attributed to the model's use of a weighted averaging ensemble method, effectively combining the prediction results of multiple models to enhance overall performance. YOLOX-Tiny and Faster RCNN are both popular object detection models currently in use. YOLOX-Tiny performs well in simple scenarios but sees a drop in performance in complex ones, due to the limitations of the YOLOX series models in dealing with small targets and complex backgrounds. Faster RCNN performs better in close and distant scenarios but has a lower average precision in complex scenarios, which may be because the RCNN series models are not as fast as the YOLO series models and are relatively weaker in adapting to complex backgrounds. YOLOv3-Tiny and SSD have relatively lower average precision across all scenarios, but SSD achieves an FPS of 94.6. This may be because the SSD model employs a multi-scale feature fusion strategy, which, although increasing the computational load, is beneficial for handling targets of different sizes, thus giving it an advantage in processing speed. In addition, Improved YOLOv4-Tiny achieves relatively high average precision in all scenarios, especially outstanding in close and simple scenarios, and slightly lower in complex scenarios compared to the paper model, but still with a MAP of 86.8% and an FPS of 88.4, showing that it maintains high accuracy while also having good processing speed. The C2Net-YOLOv5 model also shows high average precision in all scenarios, with performance close to Improved YOLOv4-Tiny in close and complex scenarios, and slightly better in moderate scenarios, with a MAP of 86.6% and an FPS of 85.8. This result may reflect the good balance of performance of C2Net-YOLOv5 in different scenarios, especially in handling complex scenes. Moreover, this model's FPS is slightly lower than that of Improved YOLOv4-Tiny, but it is still at a high level, indicating that it is also effective in real-time processing. Overall, although the "Improved YOLOv4-Tiny" and "C2Net-YOLOv5" models approach the performance of our fusion model in some performance indicators, our model still shows the best balance of performance when considering different scenarios comprehensively.

Table 2. The performance of the paper model compared to other algorithms on the CCTSDB dataset.

Model	**AP50/%**					**MAP/%**	**FPS/(Frame·s^{-1})**
	Close	**Distant**	**Simple**	**Modest**	**Complex**		
Paper model	93.1	85.7	93.8	88.6	86.1	88.5	90.7
Weighted averaging ensemble model	91.2	85.3	91.3	88.3	84.5	85.4	87.6
YOLOX-Tiny	90.4	85.0	91.4	89.2	85.3	85.7	83.9
Faster RCNN	90.0	86.1	92.2	86.8	83.2	84.5	94.1
YOLOv3-Tiny	89.7	83.3	90.2	86.4	83.8	82.5	88.3
SSD	89.1	83.5	91.0	87.2	82.7	82.3	94.6
Improved YOLOv4-Tiny	91.7	85.4	92.4	89.2	85.5	86.8	88.4
C2Net-YOLOv5	91.4	85.6	92.3	88.9	86.1	86.6	85.8

Figure 5 shows the loss function curves for detecting targets at close and distant ranges. As evident from Figure 5, with increasing training iterations, the loss function curves of all algorithms tend to stabilize without showing signs of overfitting. In the detection of distant targets, YOLOX-Tiny and YOLOv3-Tiny exhibit significant fluctuations and higher maximum values in their loss function curves, indicating a generally unstable detection capability of the YOLO model for distant targets. YOLOv3-Tiny is less efficient overall than YOLOX-Tiny, due to the latter's introduction of an anchor-free model compared to YOLOv3. When detecting close targets, the differences in the loss function curves of the algorithms are negligible. However, the advantages of the algorithm proposed in this paper are clear in the detection of distant targets, showing a significant improvement over the initially weaker detection capabilities of YOLOv3-Tiny and YOLOX-Tiny for distant targets. Compared to SSD and Faster RCNN, the loss function curve of the proposed algorithm is also smoother. Coupled with the results from Table 2, the stacking ensemble model requires less computational effort and offers superior detection performance, balancing the needs for detection speed and accuracy. Unlike common methods that trade off some detection speed for higher accuracy, such as bootstrapping aggregation, the stacking ensemble algorithm based on the weighted average method is more suited for detecting targets in the peripheral vision of drivers on highways and in densely populated areas, contributing to enhanced driving safety.

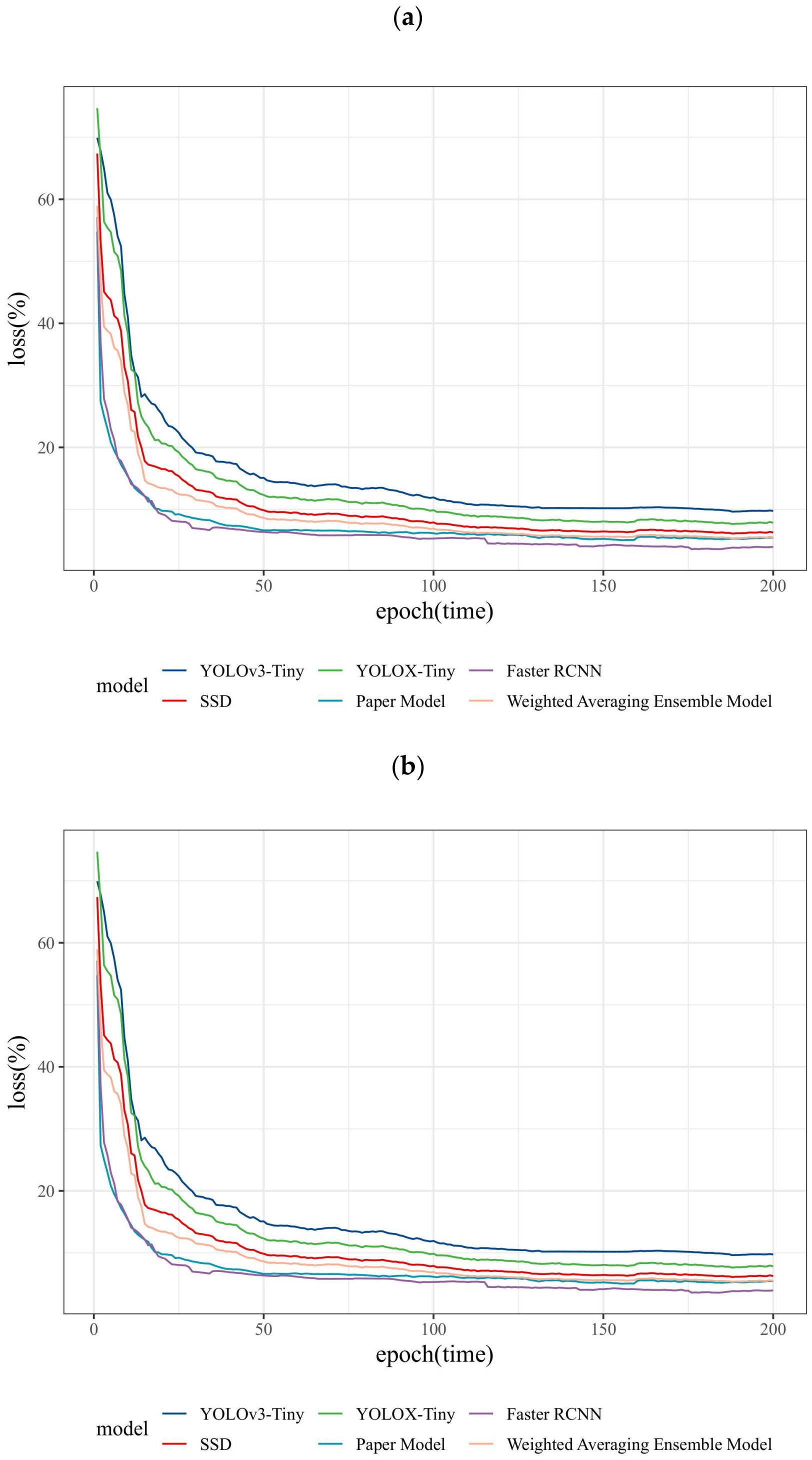


Figure 5. Loss function curves for near (**a**) and far (**b**) traffic sign detection in CCTSDB dataset.

4.2. Results on MTSD

To ensure the generalizability and universality of the methods presented in this paper, we also selected the Mapillary Traffic Sign Dataset (MTSD) dataset, which includes traffic signs from a global spectrum, for experimentation. The fully annotated set of the MTSD includes a total of 52,453 images with 257,543 traffic sign bounding boxes. The additional, partially annotated dataset contains 47,547 images with more than 80,000 signs that are automatically labeled with correspondence information from 3D reconstruction. Figure 6 illustrates the traffic signs of various categories within the MTSD dataset. We have trained our model for 28,000 iterations on the dataset. The dataset was divided into training and testing sets with a ratio of 8:2. The training platform used was Python 3.8, and the graphics card was an NVIDIA GTX 3060 12G.

Figure 6. Traffic signs of various categories within the MTSD dataset.

Utilizing the MTSD dataset, which encompasses a broader range of shapes and colors within each traffic sign category, all models generally exhibited lower mAP values on MTSD compared to the CCTSDB dataset. In contrast to their performance on the CCTSDB dataset, our model demonstrated superior performance on MTSD. This dataset's inclusion of a greater number of targets classified into varied categories led to a diversification of labels for identical target objects, complicating the model's ability to learn features within a single category. For instance, the Regulatory category in the MTSD dataset comprises 176 distinct target objects, and the Complementary category includes 45 different targets, with the Warning and Information categories also containing a multitude of varied objects.

Table 3 provides a performance comparison between our proposed fusion model and several classic traffic sign detection and recognition algorithms. It can be observed from the table that our paper model outperforms other models in terms of Precision, Recall, and mean Average Precision (MAP), with a MAP of 87.6%, indicating a significant advantage in recognition accuracy. Additionally, the model's FPS is 85.2, demonstrating good real-time processing capabilities. Other models such as YOLOX-Tiny, Faster RCNN, YOLOv3-Tiny, and SSD also perform well but still lag behind our paper model. YOLOX-Tiny's MAP is 82.5%, and Faster RCNN's is 83.6%, both lower than our model. YOLOv3-Tiny and SSD also fall behind our model in terms of Precision and Recall, although SSD's FPS has reached 85.9, showing a faster processing speed. It is worth noting that the "Improved YOLOv4-Tiny" and "C2Net-YOLOv5" models also show strong competitiveness in some aspects. "Improved YOLOv4-Tiny" scored 84.0% in Precision and 82.4% in Recall, with a MAP of 86.6%, and an FPS of 85.0, showing performance close to our model. "C2Net-YOLOv5" performed the best among all other models, with Precision and Recall of 83.9% and 82.7% respectively, a MAP as high as 86.9%, and an FPS of up to 93.6, indicating its potential advantages in both recognition accuracy and processing speed. Overall, although the "Improved YOLOv4-Tiny" and "C2Net-YOLOv5" models approach the performance of our fusion model in some performance indicators, our model still shows the best balance of performance when considering Precision, Recall, and processing speed comprehensively. These results indicate that our fusion model not only performs best in accuracy but also has strong competitiveness in real-time processing capabilities, making it an ideal choice for traffic sign detection tasks.

Table 3. The performance of the paper model compared to other algorithms on the MTSD dataset.

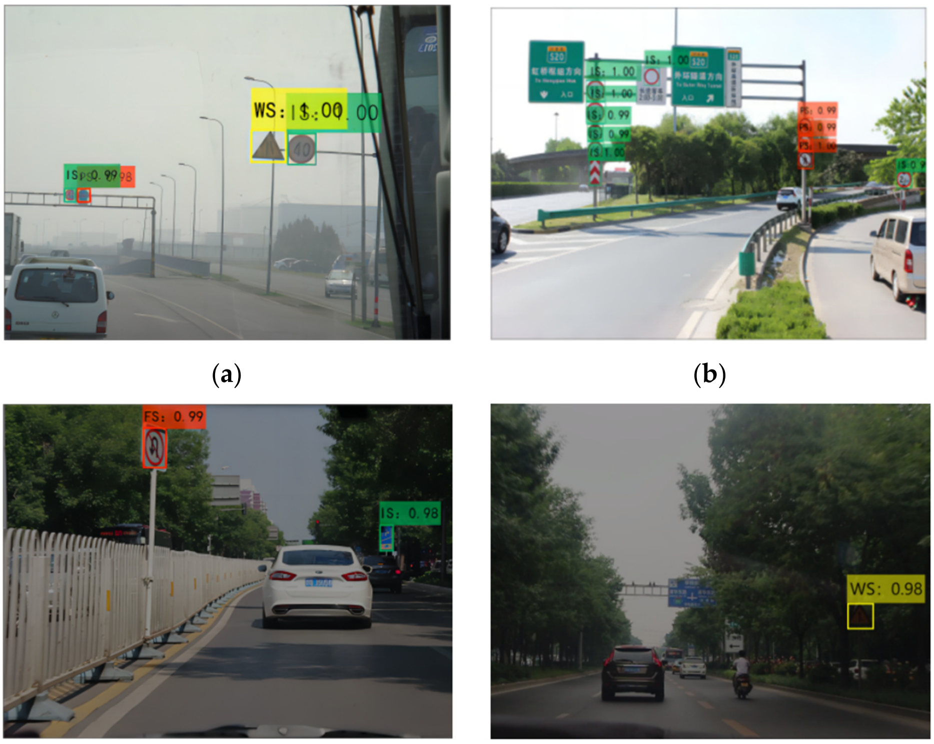
Model	Precision/%	Recall/%	MAP/%	FPS/(Frame·s^{-1})
Paper model	85.5	83.3	87.6	85.2
Weighted averaging ensemble model	82.9	81.8	84.3	86.3

YOLOX-Tiny	82.6	82.4	82.5	80.9
Faster RCNN	83.3	82.6	83.6	85.7
YOLOv3-Tiny	78.6	80.4	78.9	81.3
SSD	78.2	79.2	79.0	85.9
Improved YOLOv4-Tiny	84.0	82.4	86.6	85.0
C2Net-YOLOv5	83.9	82.7	86.9	93.6

4.3. Qualitative Analysis

In this section, we conducted a qualitative analysis of the images output by our model across various categories and scenes, showcasing the performance of our model on these images.

Initially, we presented the results of our model on the CCTSDB dataset images, followed by its performance on the MTSD dataset. Figure 7a depicts images of traffic signs detected from a considerable distance on a vehicle, which are not clearly visible to the naked eye, yet our model has accurately detected and recognized these signs with 99% and 98% confidence. Figure 7b shows images of traffic signs located on a highway. A variety of traffic signs are present in a single location, but our model has successfully detected all traffic signs and accurately labeled each one, including various indication signs (ISs) and prohibition signs (PSs), with all road traffic signs detected with over 99% confidence. Figure 7c illustrates a scenario with a partially obscured traffic sign. Our model successfully detected the indication sign (IS) with 98% confidence, despite it being only partially visible. Moreover, Figure 7d displays traffic signs under various lighting conditions, where the signs are shrouded in shadow and exhibit lower visibility compared to daylight conditions. Our model identified the warning sign (WS) with 98% confidence, demonstrating the robustness of our model under varying light conditions.

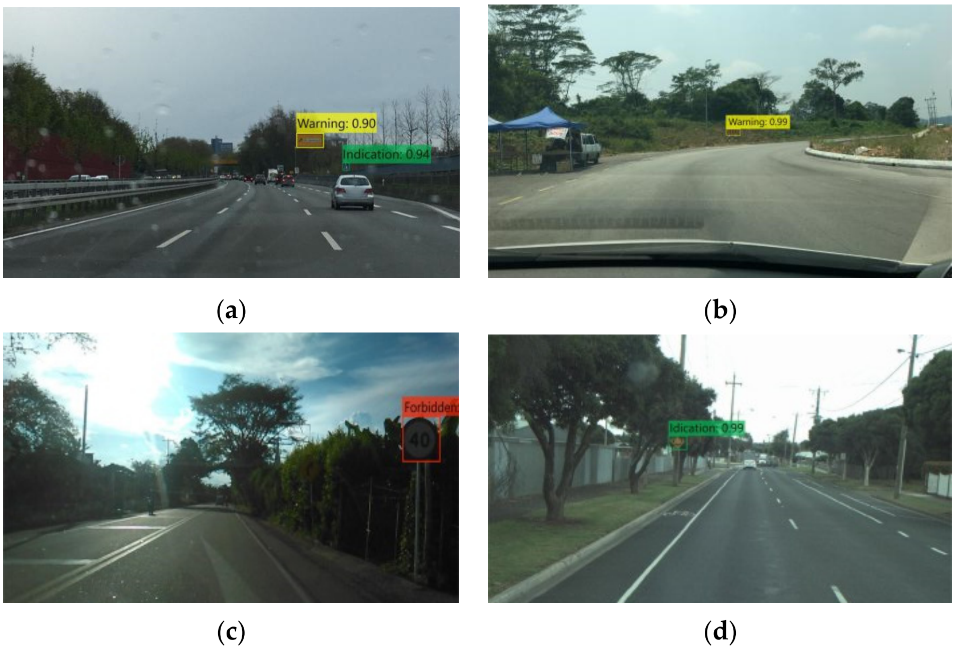


(**a**) (**b**)

(**c**) (**d**)

Figure 7. Recognition by paper model in various complex scenarios within the CCTSDB dataset.

Furthermore, we performed a qualitative analysis on images from the MTSD dataset. The MTSD dataset comprises images of varied resolution and includes more challenging images under different conditions. It is a global dataset containing signs in various languages. Figure 8a shows an image of a traffic sign during rain, where raindrops are visible on the lens, yet our model correctly predicted the warning sign and indication sign with 90% and 94% confidence, respectively. Under adverse weather conditions, drivers may not clearly see the different traffic signs due to low visibility. Motion blur, caused by movement, also presents a challenge in detecting signs while driving. Figure 8b displays a case where a warning sign, which closely resembles the color of the surrounding soil, is still successfully recognized by our model. Figure 8c demonstrates that, in a strong light environment, a No Speeding Over 40 prohibition sign completely covered by shadows is still successfully recognized by our fusion model. Figure 8d illustrates the capability of our model to recognize traffic signs partially obscured by the shadows of leaves, successfully detecting the indication sign with 99% confidence. The experiments indicate that our model is capable of accounting for these scenarios and performs better accordingly.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Recognition by paper model in various complex scenarios within the MTSD dataset.

4.4. Ablation Experiments

In this section, we examine the capability of our neural network to recognize traffic signs in the test set that were not previously learned during training, from the perspective of computer vision. Initially, we removed the right-hand curve and left hand curve signs from the MTSD dataset and trained the model on the modified MTSD dataset; the experiments were conducted using Python 3.8 on the same system equipped with an NVIDIA GTX 3060 12G graphics card. The results demonstrated that all right-hand curve and left-hand curve signs were recognized as Y-intersections. From Figure 9, it can be observed that all warning category signs feature a triangular warning symbol, and signs indicating left-hand curve, right-hand curve, and Y-intersections bear a certain resemblance. Therefore, we have reason to suspect that our model is capable of recognizing parts of the signs in untrained images that have been identified.

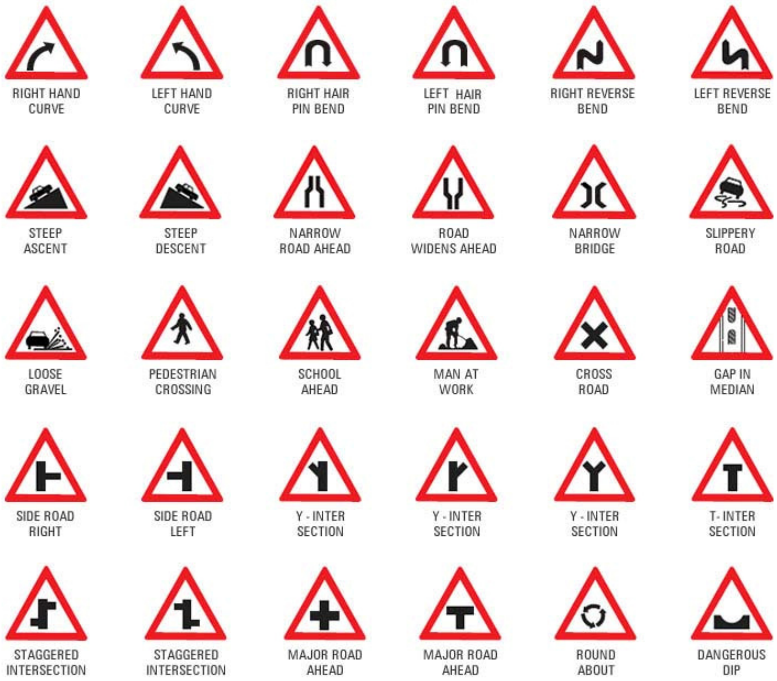
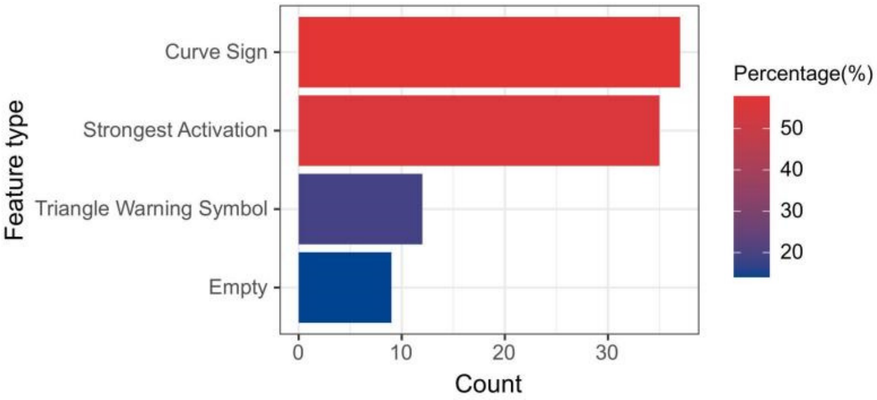
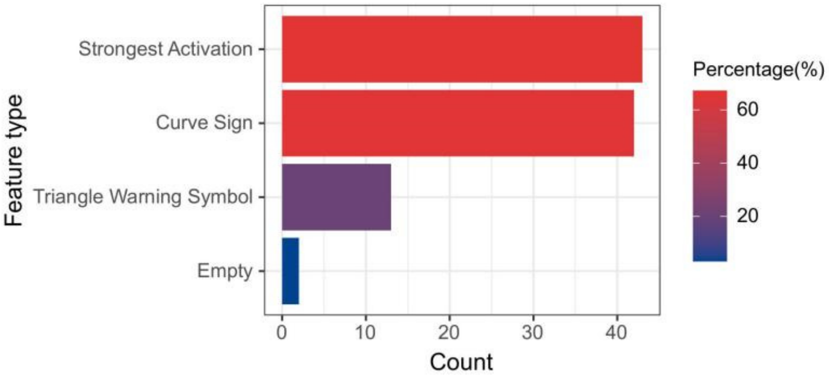


Figure 9. Comprehensive chart of various warning category signs.

To investigate the cause of this phenomenon, we analyzed the activation strength in the convolutional layers. We observed the first convolutional layer, as this layer extracts the simplest patterns of the deep network. The deeper the layer, the more complex and difficult to understand the patterns become. The information is summarized in the bar chart presented in Figure 10. Observing the first convolutional layer, we discovered that only 57% of the channels extracted the curve signs, 18.6% included the triangle warning symbol, 14.5% were empty, and 58.3% had the strongest activation. Thus, 75.6% of the channels extracted at least one warning symbol or curve signs. This explains why these signs were ultimately classified as Y-intersections. On the other hand, all Y-intersection scenarios were correctly classified. From Figure 10b, it was observed that 64.65% of the channels extracted the intersection sign, 17.68% extracted the warning symbol, 3.63% of the channels were empty, and 66.73% had the strongest activation.



(**a**)



(**b**)

Figure 10. Statistical chart of model feature extraction for curve (**a**) and Y-intersection (**b**) scenarios.

The initial ablation study indicated that the activation channel's intensity diminishes when our neural network is presented with a sign it has not encountered during training, resulting in a higher prevalence of inactive channels in the primary convolutional layer. This deactivation occurs as the left-hand curve, right-hand curve, and Y-intersection signs share a common element: the warning symbol. Consequently, our stacking ensemble model YOLOX-Tiny-RCNN demonstrates robustness by being able to recognize components of traffic signs that were not part of its training regimen.

The subsequent segment of the ablation research focused on assessing the impact of batch size on the training of the network. Each network underwent training utilizing the hyperparameters delineated in Table 4 on the CCTSDB dataset. The investigation involved modifying the batch size to 64, 128, and 256, respectively.

Table 4. Summary of hyperparameters used on the CCTSDB dataset.

Hyperparameters	Paper Model
Learning rate	0.001
Factor for dropping the learning rate	0.1
Number of epochs for dropping the learning rate	8
Maximum number of epochs	40
Momentem	0.9
Shuffle	Once

Figure 11 describes the mean Average Precision (mAP) performance across different batch sizes for paper's model and two classical models. For the Faster RCNN model, mean Average Precisions (mAPs) of 79.96%, 82.90%, and 83.61% were realized for batch sizes of 64, 128, and 256, respectively, indicating an enhancement in network performance with an increase in batch size. With regard to the YOLO X-Tiny model, a mAP of 82.65% was recorded for both batch sizes of 64 and 128. A slight increment to 83.60% was observed when the batch size was augmented to 256, highlighting a minimal network improvement as the batch size expanded. Subsequently, the stacking ensemble model presented in this study was evaluated. Accuracies of 84.71%, 85.67%, and 84.67% were attained for batch sizes of 64, 128, and 256, respectively. It was noted that, relative to the initial batch size of 64, modifications to batch sizes 128 and 256 did not manifest significant changes in the performance of our stacking ensemble model.

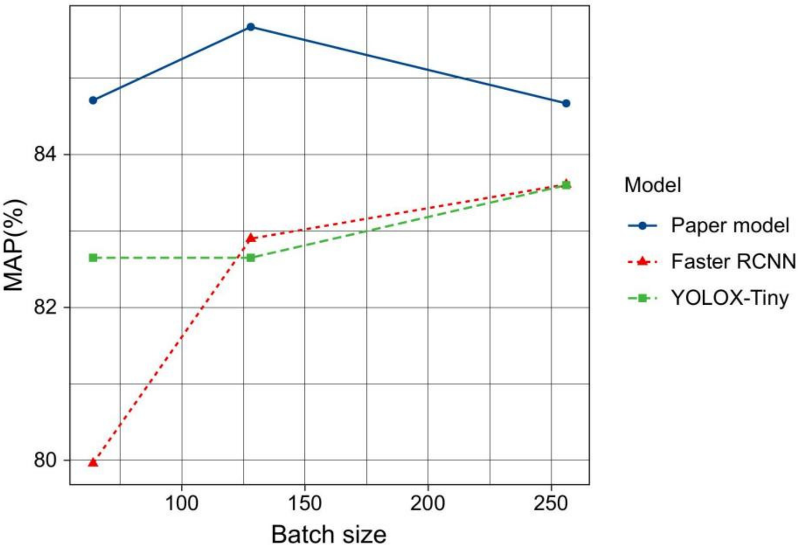


Figure 11. Mean Average Precision (mAP) performance across different batch sizes for paper's model and two classical models.

5. Conclusions

In this paper, we proposed a novel model designed to address the challenge of real-time traffic sign detection and recognition by leveraging the YOLOX-Tiny and Faster RCNN frameworks. Through the stacking ensemble process, we significantly enhanced the neural network's robustness in detecting and recognizing traffic signs. We categorized the Changsha University of Science and Technology dataset into five classes based on the content captured by the camera: complex, moderate, simple, long-range, and close-range. Extensive experiments were conducted on the proposed model, as well as several classical single-stage and two-stage models. Unprecedented mean Average Precision (mAP) scores of 94.80% and 80.71% were achieved on these datasets, respectively. Additionally, experiments were conducted on the MTSD dataset to assess the model's performance on traffic signs that encompass a broader range of colors and more complex scenarios. In subsequent quantitative analysis experiments, including scenarios such as ultra-long-distance cases, multiple traffic signs in the same scene, partial occlusion cases, shadow interference cases, raindrop interference cases, and motion blur cases, recognition accuracy of over 98% was consistently achieved.

The traffic sign recognition algorithm proposed in this paper offers valuable insights for further developments in the field, providing a more diversified model structure that enhances recognition accuracy while ensuring processing efficiency. This efficient traffic sign recognition technology is pivotal for improving the safety of autonomous vehicles, helping to reduce traffic accidents and increasing road usage efficiency, thereby promoting the sustainable development of transportation systems. Moreover, by accurately recognizing traffic signs, traffic flow management can be optimized, reducing congestion, and lowering energy consumption and emissions, which is significant for achieving an environmentally friendly transportation system.

In future work, we plan to conduct experiments on the ITSD dataset and include more challenging images for training and testing with diverse models. Furthermore, we aim to focus on integrating a model capable of effectively processing challenging images, such as those that are blurry, partially visible, or of poor quality. These improvements will further enhance the model's applicability under real-world conditions and contribute to the realization of smarter, safer, and more environmentally friendly transportation systems. Through these efforts, we expect to provide strong technical support for the sustainable development of future transportation.

Author Contributions: Y.W. conceived the experiment(s), Y.W. and Q.W. conducted the experiment(s), Q.W. analyzed the results, J.W. contributed to the data collection and processing. All authors reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in https://github.com/csust7zhangjm/CCTSDB (accessed on 29 July 2024).

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

References

1. Macioszek, E.; Granà, A.; Fernandes, P.; Coelho, M.C. New Perspectives and Challenges in Traffic and Transportation Engineering Supporting Energy Saving in Smart Cities—A Multidisciplinary Approach to a Global Problem. *Energies* **2022**, *15*, 4191.
2. Tumminello, M.L.; Macioszek, E.; Granà A. Insights into Simulated Smart Mobility on Roundabouts: Achievements, Lessons Learned, and Steps Ahead. *Sustainability* **2024**, *16*, 4079.
3. Luo, H.; Chen, H. Survey of Object Detection Based on Deep Learning. *Acta Electron. Sin.* **2020**, *48*, 1230–1239.
4. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Key Points. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
6. Lienhart, R.; Maydt, J. An extended set of Haar-like Features for Rapid Object Detection. In Proceedings of the: International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; pp. 901–904.
7. Shawe-Tylor, J.; Cristianini, N. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000; Volume 30, pp. 103–115.
8. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm, In Proceedings of the: International Conference on Machine Learning, Garda, Italy, 28 June–1 July,1996; pp. 148–156.
9. Liaw, A.; Wiener, M. Classification and Regression by Random-Forest. *R News* **2002**, *2*, 18–22.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, Realtime Object Detection, Unified, Real-Time Object Detection, May. *arXiv* **2016**, arXiv:1506.02640.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the NIPS, Montreal, Canada, 7–10 December 2015; pp. 91–99.
13. Zhang, J.; Huang, M.; Jin, X.; Li, X. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2. *Algorithms* **2017**, *10*, 127–140.
14. Ertler, C.; Mislej, J.; Ollmann, T.; Porzi, L.; Neuhold, G.; Kuang, Y. *The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale*; Springer International Publishing: Cham, Switzerland, 2020; pp. 68–84. https://doi.org/10.1007/978-3-030-58592-1_5.
15. 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: European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
16. Wu, S.; Wong, H. Joint Segmentation of Collectively Moving Objects Using a Bag-of-Words Model and Level Set Evolution. *Pattern Recognit.* **2012**, *45*, 3389–3401.
17. Wang, C.; Li, D.; Li, Z.; Wang, D.; Dey, N.; Biswas, A.; Moraru, L.; Sherratt, R.; Shi, F. An Efficient Local Binary Pattern Based Plantar Pressure Optical Sensor Image Classification Using Convolutional Neural Networks. *Optik* **2019**, *185*, 543–557.
18. Zhao, R.; Tang, S.H.; Shen, J.; Supeni, E.E.B.S.; Rahim, S.A. Enhancing Autonomous Driving Safety: A robust Traffic Sign Detection and Recognition Model TSD-YOLO. *Signal Process.* **2024**, *225*, 109619.
19. Megalingam, R.K.; Thanigundala, K.; Musani, S.R.; Nidamanuru, H.; Gadde, L. Indian Traffic Sign Detection and Recognition Using Deep Learning. *Int. J. Transp. Sci. Technol.* **2023**, *12*, 683–699.
20. Rani, A.; Anusha, Y.; Cherishama, S.K.; Vijaya Laxmi, S. Traffic Sign Detection and Recognition Using Deep Learning-Based Approach with Haze Removal for Autonomous Vehicle Navigation. e-Prime–Advances in Electrical Engineering. *Electron. Energy* **2024**, *7*, 100442.
21. Ren, B.; Zhang, J.; Wang, T. A Hybrid Feature Fusion Traffic Sign Detection Algorithm Based on YOLOv7. Computers. *Mater. Contin.* **2024**, *80*, 1425–1440.
22. Kedkarn, C.; Anusara, H.; Ratiporn, C.; Kittisak, K.; Nittaya, K. Traffic Sign Classification Using Support Vector Machine and Image Segmentation. In Proceedings of the: International Conference on Industrial Application and Engineering, Phuket, Thailand, 20–21 December 2015; pp. 52–58.
23. Kumar, A.; Zhang ZuoPeng, J.; Lyu, H. Object Detection in Real Time Based on Improved Single Shot Multi-Box Detector Algorithm. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 204.

24. Tabernik, D.; Skoaj, D. Deep Learning for Large-Scale Traffic-Sign Detection and Recognition. *IEEE Trans. Intel. Trans. Syst*. **2020**, *21*, 1427–1440.
25. Li, C.J.; Qu, Z.; Wang, S.Y.; Liu, L. A Method of Cross-layer Fusion Multi-object Detection and Recognition Based on Improved Faster R-CNN Model in Complex Traffic Environment. *Pattern Recognit. Lett*. **2021**, *145*, 127–134.
26. Carrasco, D.P.; Rashwan, H.A.; García, M.Á.; Puig, D. T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks. *IEEE Access* **2023**, *11*, 22430–22440.
27. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks Detection in IoT-Based Smart City Applications Using Machine Learning Techniques. *Int. J. Environ. Res. Public Health* **2020**, *17*, 9347.
28. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised Machine Learning: A Review of Classification Techniques. *Emerg. Artif. Intell. Appl. Comput. Eng*. **2007**, *160*, 3–24.
29. Webb, G.I.; Zheng, Z. Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. *IEEE Trans. Knowl. Data Eng*. **2004**, *16*, 980–991.
30. Aburomman, A.A.; Reaz, M.B.I. A novel SVM-KNN-PSO Ensemble Method for Intrusion Detection *System. Appl. Soft Comput.* **2016**, *38*, 360–372.
31. Tama, B.A.; Rhee, K.H. Performance Evaluation of Intrusion Detection System Using Classifier Ensembles. *Int. J. Internet Protoc. Technol.* **2017**, *10*, 22–29.
32. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247.
33. Sarmas, E.; Spiliotis, E.; Marinakis, V.; Koutselis, T.; Doukas, H. A Meta-Learning Classification Model for Supporting Decisions on Energy Efficiency Investments. *Energy Build*. **2022**, *258*, 111836.
34. Zhang, H.; Li, J.-L.; Liu, X.-M.; Dong, C. Multi-Dimensional Feature Fusion and Stacking Ensemble Mechanism for Network Intrusion Detection. *Future Gener. Comput. Syst.* **2021**, *122*, 130–143.
35. Bai, Y.; Guo, Y.; Zhang, Q.; Cao, B.; Zhang, B. Multi-Network Fusion Algorithm with Transfer Learning for Green Cucumber Segmentation and Recognition Under Complex Natural Environment. *Comput. Electron. Agric.* **2022**, *194*, 106789. https://doi.org/10.1016/j. compag.2022.106789.
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Brown, I.; Mues, C. An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets. *Expert Syst. Appl.* **2012**, *39*, 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033.
38. Yao, Y.; Han, L.; Du, C.; Xu, X.; Jiang, X. Traffic Sign Detection Algorithm Based on Improved YOLOv4-Tiny. *Signal Process. Image Commun.* **2022**, *107*, 116783.

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.