axioms

MDPI

Article

Two Novel Models for Traffic Sign Detection Based on YOLOv5s

Wei Bai [1], Jingyi Zhao [1], Chenxu Dai [1], Haiyang Zhang [2], Li Zhao [3], Zhanlin Ji [1,4,*] and Ivan Ganchev [4,5,6,*]

[1] College of Artificial Intelligence, North China University of Science and Technology, Tangshan 063210, China
[2] Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China
[3] Research Institute of Information Technology, Tsinghua University, Beijing 100080, China
[4] Telecommunications Research Centre (TRC), University of Limerick, V94 T9PX Limerick, Ireland
[5] Department of Computer Systems, University of Plovdiv "Paisii Hilendarski", 4000 Plovdiv, Bulgaria
[6] Institute of Mathematics and Informatics—Bulgarian Academy of Sciences, 1040 Sofia, Bulgaria
* Correspondence: zhanlin.ji@ncst.edu.cn (Z.J.); ivan.ganchev@ul.ie (I.G.)

Abstract: Object detection and image recognition are some of the most significant and challenging branches in the field of computer vision. The prosperous development of unmanned driving technology has made the detection and recognition of traffic signs crucial. Affected by diverse factors such as light, the presence of small objects, and complicated backgrounds, the results of traditional traffic sign detection technology are not satisfactory. To solve this problem, this paper proposes two novel traffic sign detection models, called YOLOv5-DH and YOLOv5-TDHSA, based on the YOLOv5s model with the following improvements (YOLOv5-DH uses only the second improvement): (1) replacing the last layer of the 'Conv + Batch Normalization + SiLU' (CBS) structure in the YOLOv5s backbone with a **t**ransformer self-attention module (T in the YOLOv5-TDHSA's name), and also adding a similar module to the last layer of its neck, so that the image information can be used more comprehensively, (2) replacing the YOLOv5s coupled head with a **d**ecoupled **h**ead (DH in both models' names) so as to increase the detection accuracy and speed up the convergence, and (3) adding a **s**mall-object detection layer (S in the YOLOv5-TDHSA's name) and an **a**daptive anchor (A in the YOLOv5-TDHSA's name) to the YOLOv5s neck to improve the detection of small objects. Based on experiments conducted on two public datasets, it is demonstrated that both proposed models perform better than the original YOLOv5s model and three other state-of-the-art models (Faster R-CNN, YOLOv4-Tiny, and YOLOv5n) in terms of the mean accuracy (*mAP*) and *F1 score*, achieving *mAP* values of 77.9% and 83.4% and *F1 score* values of 0.767 and 0.811 on the TT100K dataset, and *mAP* values of 68.1% and 69.8% and *F1 score* values of 0.71 and 0.72 on the CCTSDB2021 dataset, respectively, for YOLOv5-DH and YOLOv5-TDHSA. This was achieved, however, at the expense of both proposed models having a bigger size, greater number of parameters, and slower processing speed than YOLOv5s, YOLOv4-Tiny and YOLOv5n, surpassing only Faster R-CNN in this regard. The results also confirmed that the incorporation of the T and SA improvements into YOLOv5s leads to further enhancement, represented by the YOLOv5-TDHSA model, which is superior to the other proposed model, YOLOv5-DH, which avails of only one YOLOv5s improvement (i.e., DH).

Keywords: computer vision; object detection; traffic sign detection; you only look once (YOLO); attention mechanism; feature fusion

MSC: 68W01; 68T01

Citation: Bai, W.; Zhao, J.; Dai, C.; Zhang, H.; Zhao, L.; Ji, Z.; Ganchev, I. Two Novel Models for Traffic Sign Detection Based on YOLOv5s. *Axioms* **2023**, *12*, 160. https://doi.org/10.3390/axioms12020160

Academic Editor: Oscar Humberto Montiel Ross

Received: 28 December 2022
Revised: 29 January 2023
Accepted: 31 January 2023
Published: 3 February 2023

1. Introduction

The detection and recognition of traffic signs play essential roles in the fields of assisted driving and automatic driving. Traffic signs are not only the main sources for drivers to obtain the necessary road information, but they also help adjust and maintain traffic flows [1]. However, in real-life scenarios, the influence of complex weather conditions and

the existence of various categories of objects presented on the road—with a large proportion of these being small objects—have brought great challenges to the research on automatic detection and recognition of traffic signs.

There were two traffic sign detection and recognition techniques in the early days—one based on color features and the other based on shape features. Later, hybrid techniques emerged, e.g., [2], which considered both the color and geometric information of traffic signs during the feature extraction. The noise reduction and morphological processing made it easier to process images based on shapes, using the geometric information of a triangle, a circle, or a square commonly found in traffic signs, along with the RGB color information, in order to identify the images containing traffic signs. Although such a technique can detect the presence of traffic signs in images, it cannot distinguish between different classes of traffic signs.

With the emergence of deep learning, some models based on it have been applied for image classification and object detection, showing excellent performance, such as the two-stage detectors, represented by, e.g., the region-based convolutional neural networks (R-CNNs), and the single-stage detectors, represented by, e.g., You Only Look Once (YOLO) versions. R-CNN [3] was the first model applying convolutional neural networks (CNNs) for object detection. R-CNN generates candidate boxes first before detection to reduce the information redundancy, thus improving the detection speed. However, it zooms and crops images, resulting in a loss of original information. SPP-net [4] defined a spatial pyramid pooling (SPP) layer in front of the fully connected layer, which allowed one to input images of an arbitrary size and scale, thus not only breaking the constraint of fixed sizes of input images but also reducing the computational redundancy. Fast R-CNN [5] changed the original string structure of R-CNN into a parallel structure and absorbed the advantages of SPP-net, which allowed it not only to accelerate the object detection but also to improve the detection accuracy. However, if a large number of invalid candidate regions is generated, it would lead to a waste of computing power, whereas a small number of candidate regions would result in missed detection. Based on the above problems, Ren et al. proposed the concept of region proposal networks (RPNs) [6], which generates candidate regions through neural networks to solve the mismatch between the generated candidate regions and the real objects. However, these two-stage models were not superior in training and detection speed, so single-stage models, represented by the YOLO family, came into existence [7]. By creating the feature map of the input image, the learning category probability, and the boundary box coordinates of the entire image, YOLO sets the object detection as a simple regression problem. The algorithm only runs once, which of course reduces the accuracy, but allows achieving a higher processing speed than the two-stage object detectors, thus making it suitable for real-time detection of objects. The first version of YOLO, YOLOv1 [8], divides each given image into a grid system. Each grid detects objects by predicting the number of bounding boxes of the objects in the grid. However, if small objects in the image appear in clusters, the detection performance is not as sufficient. The second version, YOLOv2 [9], preprocesses the batch normalization based on the feature extraction network of DarkNet19 to improve the convergence of the network. Later, YOLOv3 [10] added logic regression to predict the score of each bounding box. It also introduced the method of Faster R-CNN giving priority to only one bounding box. As a result, YOLOv3 can detect some small objects. However, YOLOv3 cannot fit well with the ground truth. YOLOv4 [11] uses weighted real connections (WRCs), cross-mini-batch normalization (CmBN), self-adaptive training (SAT), and other methods, which allows it to not only keep suitable training and detection speed but also achieve better detection accuracy. YOLOv5 passes each batch of training data through a data loader, which performs three types of data enhancement—zooming, color space adjustment, and mosaic enhancement. From the five models produced to date based on YOLOv5, this paper proposes improvements to the YOLOv5s model, which uses two cross-stage partial connections (CSP) structures (one for the backbone network and the other for the neck) and

a weighted non-maximum suppression (NMS) [12] to improve the detection accuracy of the occluded objects in images.

The two-stage object detectors, such as R-CNN, SPP-net, and Fast R-CNN mentioned above, are not suitable for real-time detection of objects due to their relatively low detection speed. As single-stage object detectors, the YOLO versions are obviously better than the two-stage detectors in terms of the detection speed achieved. However, their detection performance is not as efficient. To tackle this problem, this paper proposes two novel YOLOv5s-based traffic sign detection models, called YOLOv5-DH and YOLOv5-TDHSA, with the following improvements to YOLOv5s (YOLOv5-DH uses only the second improvement below), which constitute the main contributions of the paper:

1. Replacing the last layer of the 'Conv + Batch Normalization + SiLU' (CBS) structure in the YOLOv5s backbone with a **t**ransformer self-attention module (T in the YOLOv5-TDHSA's name), and also adding a similar module to the last layer of its neck, so that the image information can be used more comprehensively;
2. Replacing the YOLOv5s coupled head with a **d**ecoupled **h**ead (DH in the both models' names) so as to increase the detection accuracy and speed up the convergence;
3. Adding a **s**mall-object detection layer (S in the YOLOv5-TDHSA's name) and an **a**daptive anchor (A in the YOLOv5-TDHSA's name) to the YOLOv5s neck to improve the detection of small objects.

Based on results obtained from experiments conducted on two public datasets (TT100K and CCTSDB2021), the proposed YOLOv5-DH and YOLOv5-TDHSA models outperform the original YOLOv5s model along with three other state-of-the-art models (Faster R-CNN, YOLOv4-Tiny, YOLOv5n), as shown further in the paper.

The rest of the paper is organized as follows. Section 2 introduces the attention mechanisms, feature fusion networks, and detection heads commonly used in object detection models. Section 3 presents the main representatives of the two-stage and single-stage object detection models. Section 4 explains the YOLOv5s improvements used by the proposed models, including the transformer self-attention mechanism, the decoupled head, the small-object detection layer, and the group of adaptive anchor boxes. Section 5 describes the conducted experiments, and presents and discusses the obtained results. Finally, Section 6 concludes the paper.

2. Background

2.1. Attention Mechanisms

Attention is a data processing mechanism used in machine learning and extensively applied in different types of tasks such as natural language processing (NLP), image processing, and object detection [13]. The squeeze-and-exchange (SE) attention mechanism aims to assign different weights to each feature map and focuses on more useful features [14]. SE pools the input feature map globally, then uses a full connection layer and an activation function to adjust the feature map, thus obtaining the weight of the feature, which is multiplied with the input feature at the end. The disadvantage of SE is that it only considers the channel information and ignores the spatial location information. The convolutional block attention module (CBAM) solves this problem by first generating different channel weights, and then compressing all feature maps into one feature map to calculate the weight of the spatial features [15]. Currently, the self-attention [16] is one of the most widely used attention mechanisms due to its strong feature extraction ability and the support of parallel computing. The transformer self-attention mechanism, used by the YOLOv5-TDHSA model proposed in this paper, can establish a global dependency relationship and expand the receptive field of images, thus obtaining more features of traffic signs.

2.2. Multi-Scale Feature Fusion

The feature pyramid network (FPN) [17] utilized in Faster R-CNN and Mask R-CNN [18] is shown in Figure 1a. It uses the features of the five stages of the ResNet

convolution groups C2–C6, among which C6 is obtained from a MaxPooling operation by directly applying $1 \times 1/2$ on C5. The feature maps P2–P6 are obtained after the FPN fusion, as follows: P6 is equal to C6, P5 is obtained through a 1×1 convolution followed by a 3×3 convolution, and P2–P4 are obtained through a 1×1 convolution followed by a fusion with the feature of the former $2 \times$ Upsample and a 3×3 convolution.

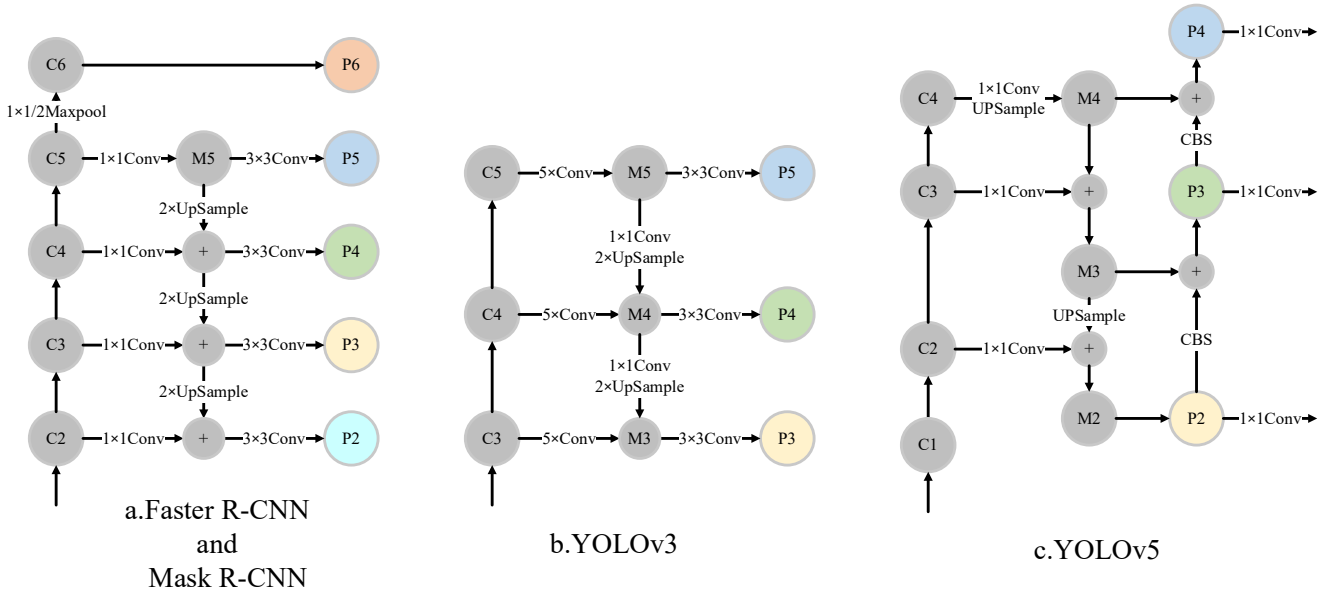

Figure 1. Different feature fusion structures.

The FPN in YOLOv3 is shown in Figure 1b. The features of C3, C4, and C5 are used. The features from C5 to P5 first pass through five layers of convolution, and then through one layer of 3×3 convolution. The features of P4 are obtained by connecting M5 (through 1×1 Conv + $2 \times$ Upsample) and C4 through five layers of convolution, and one layer of 3×3 convolution. The features of P3 are obtained by connecting M4 (through 1×1 Conv + $2 \times$ Upsample) and C3 through five layers of convolution, and one layer of 3×3 convolution.

The feature extraction network of YOLOv5 uses a 'FPN + Path Aggregation Network (PAN)' [19] structure, as shown in Figure 1c. PAN adds a bottom-up pyramid behind the FPN as a supplement. FPN conveys the strong semantic features from top to bottom, while PAN conveys strong positioning features from bottom to top. The specific operation of PAN includes first copying the last layer M2 of FPN as the lowest layer P2 of PAN, and then fusing M3 with the downsampled P2 to obtain P3. P4 is obtained through a feature fusion of M4 and downsampled P3. However, the feature extraction network does not work well for the detection of small objects. The feature fusion utilized by the YOLOv5-TDHSA model, proposed in this paper, is based on a small-object detection layer, making the detection of small objects more accurate. This is described in more detail in Section 4.3.

2.3. Detector Head

Since the head of YOLOv1 only generates two detection boxes for each grid, it is not suitable for both dense and small-object detection tasks. Its generalization ability is weak when the size ratio of the same-type objects is uncommon. The head of YOLOv2 improves the network structure and also adds an anchor box. YOLOv2 removes the last fully connected layer in YOLOv1, and uses convolution and anchor boxes to predict the detection box. However, since the use of convolution to downsample the feature map results in a loss of the fine-grained features, the model's detection of small objects is poor. Consequently, the passthrough layer structure has been introduced in the head of YOLOv2 to divide the feature map into four parts to preserve the fine-grained features. The head of

YOLOv3 introduces a multi-scale detection logic and utilizes a multi-label classification idea on the basis of YOLOv2. The loss function has been optimized as well. YOLOv4 adopts a multi-anchor strategy, different from YOLOv3. Any anchor box greater than the intersection over union (IoU) [20] threshold is regarded as a positive sample, thus ensuring that the positive samples ignored by YOLOv3 will be added to YOLOv4 to improve the detection accuracy of the model. The output of YOLOv5 has three prediction branches. The grid of each branch has three corresponding anchors. Instead of the IoU maximum matching method, YOLOv5 calculates the width–height ratio of the bounding box to the anchor of the current layer. If the ratio is greater than the parameter value set, this indicates that the matching degree is poor, which is considered as a background. The coupled detection head of YOLOv5s performs both the recognition and positioning tasks on a feature map simultaneously. However, these tasks have different focuses, making the final recognition accuracy low. The 'decoupled head' idea allows one to separate these two tasks and achieve better performance. Therefore, the models proposed in this paper use a decoupled head instead of the original YOLOv5s coupled head, which is described in more detail in Section 4.2.

3. Related Work

Over the past 20 years, the object detection models were divided into two categories: (1) traditional models (before 2012), such as V-J detection [21,22], HOG detection [23], DPM [24], etc., and (2) deep learning (DL) models, beginning with AlexNet [25]. The following subsections briefly present the DL object detection models, divided into two-stage and one-stage models, whose development route is illustrated in Figure 2.

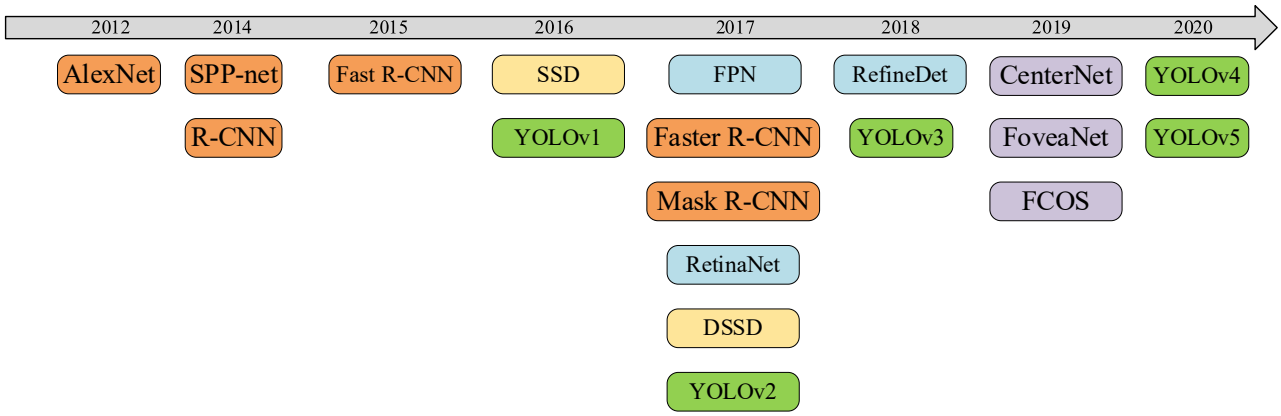

Figure 2. The development route of the DL object detection models.

3.1. Two-Stage Object Detection Models

Krizhevsky et al. proposed AlexNet as a CNN framework when participating (and winning the first place) in the ImageNet LSVRC 2012 competition. This model brought the climax to the development of deep learning.

Later, R-CNN emerged for object detection. However, R-CNN unifies the size of all candidate boxes, which causes a loss of the image content and affects the detection accuracy. Based on R-CNN, SPP-net, Fast R-CNN, Faster R-CNN, Mask R-CNN, and other models have been developed subsequently.

SPP-net was proposed in 2014. It inserts a spatial pyramid pooling layer between the CNN layer and fully connected layer, which allows it to solve the R-CNN loss of the image content caused by adjusting all candidate boxes to the same size. In order to find the location of each area in the feature map, the location information is added after the convolution layer. However, the time-consuming selective search (SS) [26] method is still used to generate the candidate areas.

On the basis of R-CNN, Fast R-CNN adds an RoI (region of interest) pooling layer and reduces the number of model parameters, thus greatly increasing the processing speed.

The method of SPP-net is used for reference, CNN is used to process the input images, and the serial structure of R-CNN is changed to a parallel structure, so that classification and regression can be carried out simultaneously, and the detection is accelerated.

In order to solve the problem that Fast R-CNN uses the SS method to generate candidate areas, Faster R-CNN uses an RPN to directly generate candidate areas, which enables the neural network to complete the detection task in an end-to-end fashion [27].

Based on Faster R-CNN, Mask R-CNN uses a fully constructive network (FCN). The model operates in two steps: (1) generating the candidate regions through an RPN, and (2) extracting the RoI features from candidate regions using RoIAlign (region of interest alignment) to obtain the probability of object categories and the location information of prediction boxes.

The two-stage object detection models are not suitable for real-time object detection because they require multiple detection and classification processes, which lowers the detection speed.

3.2. One-Stage Object Detection Models

3.2.1. YOLO

YOLO's training and detection are carried out in a separate network. The object detection is regarded as a process of solving a regression problem. As long as the input image passes through inference, the location information of the object and the probability of its category can be obtained [28]. Therefore, YOLO is particularly outstanding in terms of detection speed. There are different versions of YOLO proposed to date. Based on its fifth version, YOLOv5, five models have been produced, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5-DH and YOLOv5-TDHSA models, described in this paper, propose improvements to the YOLOv5s model, whose network structure is shown in Figure 3. A focus network structure is used at the beginning of the trunk to derive the value of every other pixel in an image. This is followed by four independent feature layers, which are stacked. At that point, the width and height information is concentrated on the channel, and the input channel is expanded four times.

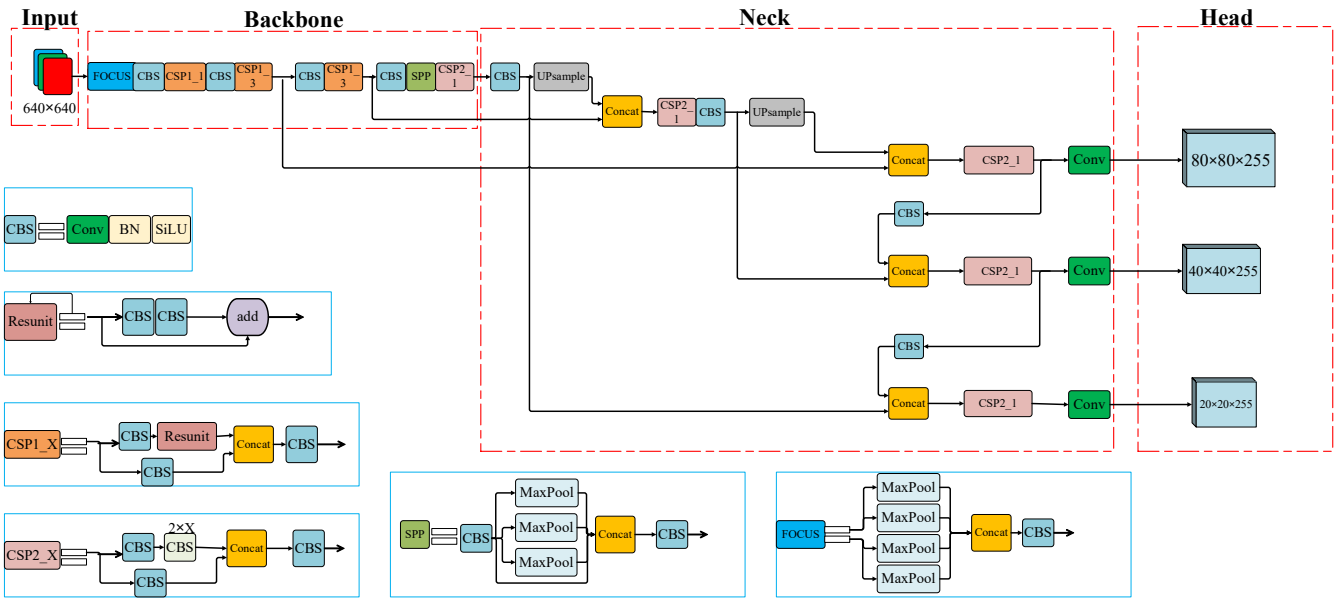

Figure 3. The structure of the YOLOv5s model.

YOLOv5s uses Mixup [29] and Mosaic for data enhancement, where Mosaic splices four images to enrich the background of the detected object. The data of the four images are processed at one time during a batch normalization computation.

In the backbone part, the model extracts features from the input image. The extracted features, through three feature layers, are used for the next network construction.

The main task of the neck part is to strengthen feature extraction and feature fusion, so as to combine feature information of different scales. In the Path Aggregation Network (PANet) structure, upsampling and downsampling operations are used to achieve feature extraction. When the input size is 640 × 640 pixels, the maximum scale of output feature is 80 × 80 pixels, so the minimum size of the detection frame is 8 × 8 pixels. However, when there are many smaller objects in the dataset, this will affect the detection accuracy. The proposed improvements of YOLOv5s in this regard are described in Section 4.3.

In the head part, the three feature layers, which have been strengthened, are regarded as a collection of feature points. This part is used to judge whether the feature points have objects corresponding to them. The YOLOv5s detection head is a coupled head which performs complete identification and location tasks on a feature map. However, recognition and location are two different tasks. Therefore, this paper proposes a branch structure to carry out recognition and location tasks separately. This improvement to the YOLOv5s structure is described in more detail in Section 4.2.

There have been some improvements of YOLOv5 recently proposed for traffic sign and traffic light recognition. For instance, Chen et al. [30] introduced a Global-CBAM attention mechanism for embedding into YOLOv5's backbone in order to enhance its feature extraction ability, and achieved sufficient balance between the channel attention and spatial attention for improving the target recognition. Due to this, the overall accuracy of the model was improved, especially for small-sized target recognition, and the mean accuracy (*mAP*) achieved was 6.68% higher than that before the improvement.

In order to solve the problem of using YOLOv5s for the recognition of small-sized traffic signs, Liu et al. [31] proposed to replace the original DarkNet-53 backbone of YOLOv5s with MobileNetV2 network for feature extraction, selecting Adam as the optimizer. The result of this was the reduction in the number of parameters by 65.6% and the computation amount by 59.1% on the basis of improving the *mAP* by 0.129.

Chen et al. [32] added additional multi-scale features to YOLOv5s to make it faster and more accurate in capturing traffic lights when these occupy a small area in images. In addition, a loop was established to update the parameters using a gradient of loss values. This led to *mAP* improvement (from 0.965 to 0.988) and detection time reduction (from 3.2 ms inference/2.5 ms to 2.4 ms inference/1.0 ms NMS per image).

3.2.2. SSD

The Single Shot MultiBox Detector (SSD) [33] is a one-stage object detection model proposed after YOLOv1. In order to improve YOLO's imperfection for small-object detection, SSD uses feature maps of different sizes and prior boxes of different sizes to further improve the regression rate and accuracy of the predicted box. The proportion of the prior frame size to the image is calculated as follows:

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m - 1}(k - 1), \quad (1)$$

where $k \in [1, m]$, m denotes the number of characteristic graphs, and S_{max} and S_{min} denote the maximum and minimum value of the ratio, respectively.

4. Proposed Improvements to YOLOv5s

This section describes the YOLOv5s improvements used by the models proposed in this paper. The decoupled head (DH) improvement is used by both proposed models, YOLOv5-DH and YOLOv5-TDHSA, whereas the other two improvements are used only by YOLOv5-TDHSA.

4.1. Transformer Self-Attention Mechanism

The transformer model was proposed by the Google team in June 2017 [34]. It has not only become the preferred model in the NLP field, but also showed strong potential in the field of image processing. The transformer abandons the sequential structure of Recurrent Neural Networks (RNNs) and adopts a self-attention mechanism to enable the model to parallelize training and make full use of the global information of training data.

The core mechanism of the transformer model is the self-attention depicted in Figure 4. The regular attention mechanism first calculates the attention distribution on all input information and then obtains the weighted average of the input information according to this attention distribution. Self-attention maps the input features to three new spaces for representation, namely Query (Q), Key (K), and Value (V). The correlation between Q and K is calculated as well, after which a *SoftMax* function is used to normalize the data and widen the gap between the data to enhance the attention. The weight coefficient and V are weighted and summed to obtain the attention value. The self-attention mechanism maps the features to three spatial representations, which allows one to avoid problems encountered when features are mapped to only one space. For example, if Q1 and Q2 are directly used to calculate the correlation, there will be no difference between the correlation between Q1 and Q2 and the correlation between Q2 and Q1. In this case, the expression ability of the attention mechanism will become weak. If K is introduced to calculate the correlation between the original data, it can reflect the difference between Q1 and K2 on one hand and Q2 and K1 on the other, which can also enhance the expression ability of the attention mechanism. Since the input of the next step is the attention weight obtained, it is not appropriate to use Q or K; thus, the third space, V, is introduced. Finally, the attention value is obtained through weighted summation.

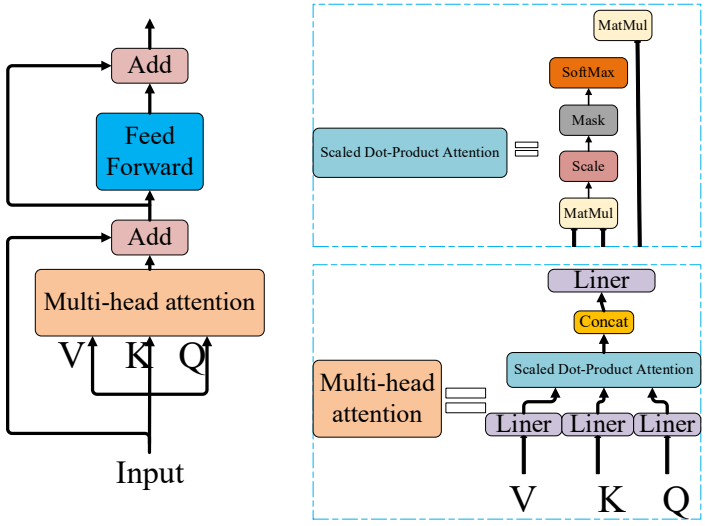

Figure 4. The module structure of the transformer self-attention mechanism.

However, the transformer model would significantly increase the amount of computation, resulting in higher training costs. The feature dimension is the smallest when the image features are transferred to the last layer of the network. At this moment, the influence on training the model would be the smallest if the transformer is added. Therefore, the proposed YOLOv5-TDHSA model uses the transformer only as a replacement of the CBS at the last layer of the backbone of the original YOLOv5s model, and also adds the transformer to the last layer of its neck.

4.2. Decoupled Head

After performing analytical experiments indicating that the coupled detection head may harm YOLO's performance, the authors of [35] recommend replacing the original YOLO's head with a decoupled one. This idea is taken on board by the models proposed in this paper to reduce the number of parameters and network depth, thus improving the model training speed and reducing the feature losses.

During the object detection, it is necessary to output the category/class and position information of the object. The decoupled head uses two different branches to output the category and position information separately as the recognition and positioning tasks have different focuses. The recognition focuses more on the existing class to which the extracted features are closer. The positioning focuses more on the location coordinates of the ground truth box so as to correct the parameters of the bounding box. YOLO's head uses a feature map to complete the two tasks of recognition and location in a convolution. Therefore, it does not perform as well as the decoupled head D1 shown in Figure 5, which is used by the models proposed in this paper. However, the decoupling process increases the number of parameters, thus affecting the training speed of the model. Therefore, in order to reduce the number of parameters, the feature first goes through a 1×1 convolution layer to reduce the dimension and then through two parallel branches with two 3×3 convolution layers. The first branch is used to predict the category. Since there are 45 categories in the TT100K dataset used in this paper, the channel dimension becomes 45 after a convolution operation and the processing of the *Sigmoid* activation function [36]. The second branch is mainly used to determine whether the object box is a foreground or background. As a result, the channel dimension becomes 1 after the convolution operation and *Sigmoid* activation function. There is also a third branch used to predict the coordinate information (x, y, w, h) of the object box. Therefore, after the convolution operation, the channel dimension becomes 4. Finally, the three outputs are integrated into $20 \times 20 \times 50$ feature information through *Concat* for the next operation. The decoupled heads D2, D3, and D4, shown in Figure 6, also follow the same steps to generate feature information of $40 \times 40 \times 50$, $80 \times 80 \times 50$, and $160 \times 160 \times 50$, respectively. The proposed YOLOv5-DH model only uses D1, D2, and D3 to replace the 'Head' part of the original YOLOv5s model (c.f., Figure 3).

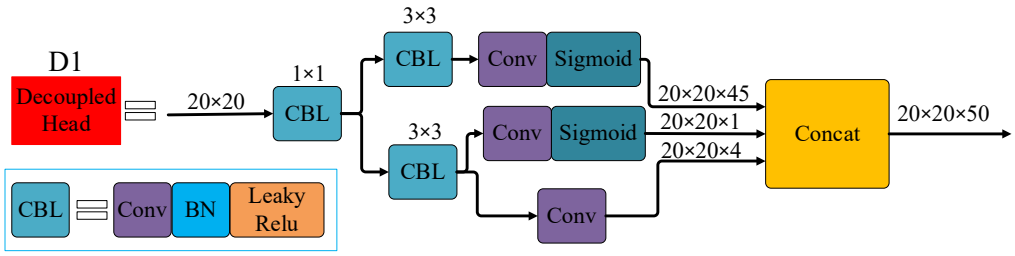

Figure 5. The structure of the decoupled head D1 used by the proposed models.

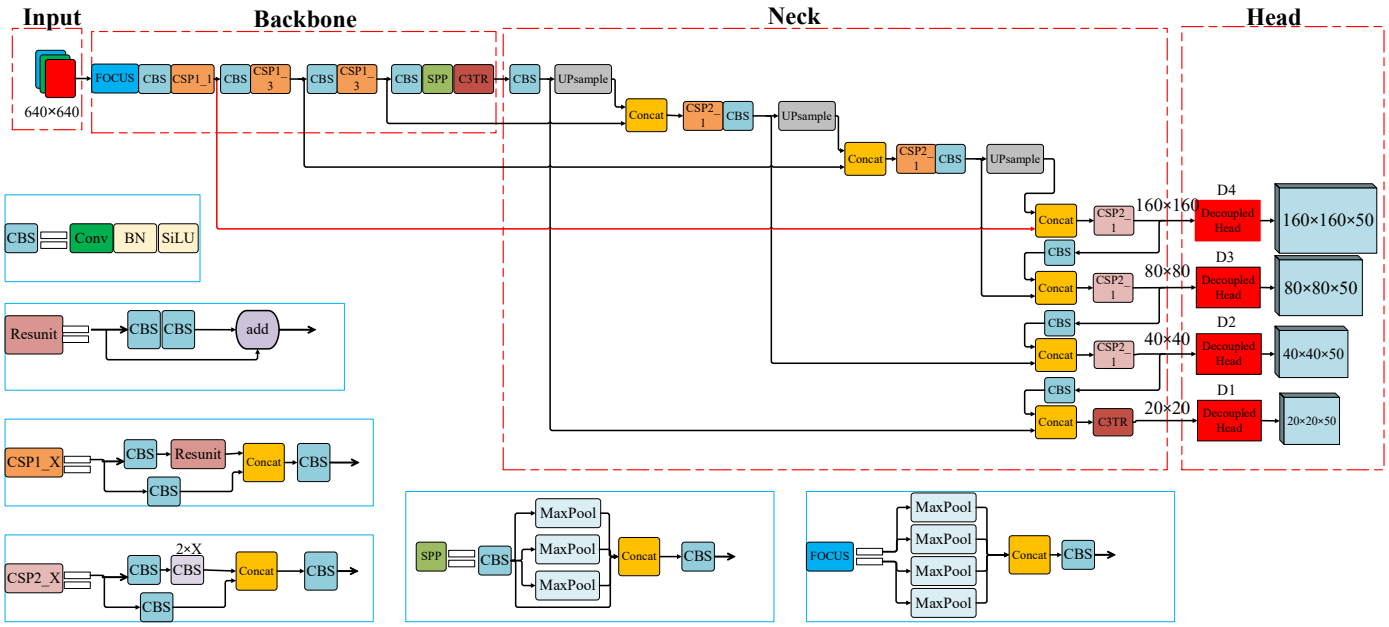

Figure 6. The structure of the proposed YOLOv5-TDHSA model.

4.3. Small-Object Detection Layer and Adaptive Anchor

During the detection of traffic signs, the changing distance between the shooting equipment and the object makes the size of traffic signs in the collected images different, which has a certain impact on the detection accuracy [37]. YOLOv5s solves this problem in the form of PANet. Taking an input image size of 640 × 640 pixels as an example, the feature information of the feature map output through the original model is 80 × 80 × 255, 40 × 40 × 255, and 20 × 20 × 255, respectively. At this time, the grid sizes of the generated detection box are 8 × 8 pixels, 16 × 16 pixels, and 32 × 32 pixels, respectively. However, when there is a large number of objects with size smaller than 8 × 8 pixels in the dataset, the detection performance for these small objects is not acceptable. Furthermore, the feature pyramid pays more attention to the extraction and optimization of the underlying features. With increasing the depth of the network, some features at the top level will be lost, reducing the accuracy of the object detection.

To improve the detection of small objects, a branch structure is added to the PANet of YOLOv5s to maintain the same size of the input image. However, the neck part adds a 160 × 160 × 128 feature information output. In other words, the feature map continues to expand by performing the convolution and upsampling on the feature map after layer 17. Meanwhile, the 160 × 160 pixels feature information obtained from layer 19 is fused with the layer 2 feature in the backbone at the layer 20 to make up for the feature loss during feature transmission. The addition of a small object detection layer in the network can ease the difficulty of small object detection. At the same time, it combines the features of the top level with those of the bottom level to supplement the features lost in the bottom level, thus improving the detection accuracy.

The network structure after the addition of the small-object detection layer is shown in Figure 6. A branch is added to connect layer 2 and layer 19 (the red solid line part). In this case, the added fourth output size is 160 × 160 × 128. After the head decoupling, the feature information size is 160 × 160 × 50. The minimum size of the generated detection box is 4 × 4 pixels, which improves the detection of small objects.

The original YOLOv5s network model has only three detection layers. As a result, there are three groups of anchor boxes corresponding to the feature maps at three different resolutions. In each group of anchor boxes, there are three different anchors. A total of nine anchors can be used to detect large, medium, and small objects. However, the YOLOv5-TDHSA model, proposed in this paper, deepens the network and adds an output layer of

feature information. It uses a group of 12 anchor boxes, added to the original YOLOv5s model, to calculate the feature map at the new resolution. The ratio between an anchor and the width and height of each ground truth box is calculated, and the K-Means and genetic learning algorithms are used to obtain the best possible recall (BPR). When BPR is greater than 0.98, it indicated that the four groups of anchor boxes generated can be suitable for custom datasets.

The addition of the small-object detection layer and the group of adaptive anchor boxes allows us to significantly improve the detection accuracy of the proposed YOLOv5-TDHSA model, as demonstrated in the next section.

5. Experiments

5.1. Datasets

Two public datasets were used in the experiments conducted for the performance comparison of models. The first one was the Tsinghua-Tencent 100 K Chinese traffic sign detection benchmark [38], denoted as TT100K in [39]. It includes 100,000 high-definition images with large variations in illuminance and weather conditions, among which 10,000 images are annotated that contain 30,000 traffic sign instances (in total), each of which theoretically belongs to one of the 221 Chinese traffic sign categories. The images are taken from the Tencent Street View Map. Sample images are shown in Figure 7. However, there is a serious imbalance in the distribution of categories in this dataset, and even some categories do not have instances corresponding to them. Therefore, in the conducted experiments, similarly to [39], only categories with more than 100 traffic sign instances were used, resulting in 45 categories spread over 9170 images.

Figure 7. Sample images of the TT100K dataset.

The other dataset used in the experiments was the CCTSDB2021 Chinese traffic sign detection benchmark [40], which was built based on the CCTSDB2017 dataset [41,42] by adding 5268 annotated images of real traffic scenes and replacing images containing easily detected traffic signs with more difficult samples of a complex and changing detection environment. Three traffic sign classes are distinguished in CCTSDB2021, namely a warning, a mandatory, and a prohibitory traffic sign class, as shown in Figure 8. There are a total of 17,856 images, including 16,356 images in the training set and 1500 images in the test set. However, the weather environment attribute, which represents a great challenge for the object detection models, is only present in the images of the test set and not of the training set. Therefore, only these 1500 images, presenting greater difficulty to the detection of traffic signs contained in them, were used in the experiments.

Figure 8. Sample images of the CCTSDB2021 dataset, containing (**A**) warning traffic signs; (**B**) mandatory traffic signs; (**C**) prohibitory traffic signs.

In the experiments, as shown in Table 1, the 9170 TT100K images and 1500 CCTSDB021 images were separately divided (using the same ratio) into a training set (60% of the total number of images), a validation set (20%), and a test set (20%). The corresponding number of labels in each of these three sets is shown in Table 1.

Table 1. Splitting the datasets into training, validation, and test sets.

	TT100K Dataset	**CCTSDB2021 Dataset**
Training set	13,908 labels	1935 labels
Validation set	4636 labels	645 labels
Test set	4636 labels	645 labels

5.2. Experimental Environment

In the training process, the initial learning rate was set to 0.01, and a cosine annealing strategy was used to reduce it. 300 epochs were performed with the batch size set to 32. The experiments were conducted on a PC with a Windows 10 operating system, Intel (R) Core (TM) i7-10,700 CPU@2.90 GHz, NVIDIA GeForce RTX3090, and 24GB video memory, by using CUDA 11.1 for training acceleration, PyTorch 1.8.1 deep learning framework for training, and an input image size of 640 × 640 pixels, as shown in Table 2.

Table 2. Experimental environment's parameters.

Component	**Name/Value**
Operating system	Windows 10
CPU	Intel (R) Core (TM) i7-10,700
GPU	GeForce RTX3090
Video memory	24 GB
Training acceleration	CUDA 11.1
Deep learning framework for training	PyTorch 1.8.1
Input image size	640 × 640 pixels
Initial learning rate	0.01
Final learning rate	0.1
Training batch size	32

5.3. Evaluation Metrics

Evaluation metrics commonly used for the performance evaluation of object detection models include *precision*, *average precision* (*AP*), *mean average precision* (*mAP*), *recall*, *F1 score*, and *processing speed* measured in frames per second (fps).

Precision refers to the proportion of the true positive (*TP*) samples in the prediction results, as follows:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

where *TP* denotes the number of images containing detected objects with IoU > 0.5, that is, the number of images containing positive samples that are correctly detected by the model; *FP* (false positive) represents the number of images containing detected objects with IoU $\leq$ 0.5.

Recall refers to the proportion of correct predictions in all positive samples, as follows:

$$recall = \frac{TP}{TP + FN}, \tag{3}$$

where *FN* (false negative) represents the number of images wrongly detected as not containing objects of interest.

The *average precision* (*AP*) is the area enclosed by the *precision–recall* curve and the X axis, calculated as follows:

$$AP = \int_0^1 p(r)dr, \tag{4}$$

where $p(r)$ denotes the precision function of recall r.

F1 score is the harmonic average of *precision* and *recall*, with a maximum value of 1 and a minimum value of 0, calculated as follows:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{5}$$

The mean average precision (*mAP*) is the mean *AP* value over all classes of objects, calculated as follows:

$$mAP = \frac{\sum AP}{N_{classes}}, \tag{6}$$

where $N_{classes}$ denotes the number of classes.

5.4. Results

Based on the two datasets, experiments were conducted for performance comparison of the proposed YOLOv5-DH and YOLOv5-TDHSA models to four state-of-the-art models, namely R-CNN, YOLOv4-Tiny, YOLOv5n, and YOLOv5s. The size and number of parameters of models are shown in Table 3 and the duration of a single experiment conducted with each model is shown in Table 4. On the two datasets, TT100K and CCTSDB2021, five separate experiments were performed with each of the models compared. In each experiment, the same data were utilized for all models, generated by randomly splitting the used dataset into a training set, a validation set, and a test set, as per Table 1. The results obtained for each model were averaged over the five experiments in order to serve as the final evaluation of the model performance.

Table 3. The size and number of parameters of compared models.

Model	Size (MB)	Number of Parameters (Million)
Faster R-CNN	360.0	28.469
YOLOv4-Tiny	22.4	6.057
YOLOv5n	3.6	1.767
YOLOv5s	13.7	7.068
YOLOv5-DH	**22.8**	**11.070**
YOLOv5-TDHSA	**24.8**	**12.224**

Table 4. Single experiment duration of compared models.

Dataset	Faster R-CNN	YOLOv4-Tiny	YOLOv5n	YOLOv5s	YOLOv5-DH	YOLOv5-TDHSA
TT100K	47 h	37.5 h	30 h	32 h	**33 h**	**35 h**
CCTSDB 2021	8.5 h	4 h	0.8 h	1 h	**2 h**	**2.5 h**

Tables 5–10 show the *mAP* and *F1 score* results obtained in each experiment, conducted on the TT100K dataset, for each of the models compared. Table 11 shows the averaged *mAP* and *F1 score* results over the five experiments, along with the processing speed achieved, measured in frames per second (fps). The obtained results, shown in Table 11, demonstrate that on the TT100K dataset, both proposed models (YOLOv5-DH and YOLOv5-TDHSA) outperform all four state-of-the-art models in terms of *mAP* and *F1 score*, at the expense of having a bigger size, greater number of parameters, and slower processing speed (surpassing only Faster R-CNN). From the two proposed models, YOLOv5-TDHSA is superior to YOLOv5-DH in terms of both evaluation metrics (*mAP* and *F1 score*).

Table 5. Results of Faster R-CNN on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	52.9	53.6	54.1	53.4	52.6
F1 score	0.576	0.581	0.586	0.579	0.575

Table 6. Results of YOLOv4-TINY on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	57.7	62.8	63.1	64.6	63.2
F1 score	0.608	0.672	0.655	0.654	0.675

Table 7. Results of YOLOv5n on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	66.0	66.2	65.1	66.3	66.6
F1 score	0.651	0.645	0.639	0.646	0.641

Table 8. Results of YOLOv5s on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	74.5	75.6	75.2	75.3	75.1
F1 score	0.728	0.741	0.740	0.730	0.728

Table 9. Results of YOLOv5-DH on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	77.2	78.3	77.6	78.5	78.1
F1 score	0.762	0.771	0.762	0.772	0.769

Table 10. Results of YOLOv5-TDHSA on TT100K dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	83.3	83.5	82.6	83.3	84.2
F1 score	0.819	0.811	0.797	0.810	0.816

Table 11. Results of compared models on TT100K dataset.

Model	*F1Score*	*mAP* (%)	Processing Speed (fps)
Faster R-CNN	0.579	53.3	40
YOLOv4-Tiny	0.653	62.3	160
YOLOv5n	0.644	66.0	111
YOLOv5s	0.733	75.1	100
YOLOv5-DH	**0.767**	**77.9**	**84**
YOLOv5-TDHSA	**0.811**	**83.4**	**77**

Tables 12–17 show the *mAP* and *F1 score* results obtained in each experiment, conducted on the CCTSDB2021 dataset, for each of the models compared. Table 18 shows the averaged *mAP* and *F1 score* results over the five experiments, along with the processing speed achieved. The obtained results, shown in Table 18, demonstrate that both proposed models (YOLOv5-DH and YOLOv5-TDHSA) outperform all four state-of-the-art models in terms of *mAP* and *F1 score* on this dataset as well, at the expense of having a bigger size, greater number of parameters, and slower processing speed (surpassing only Faster R-CNN). From the two proposed models, YOLOv5-TDHSA is again superior to YOLOv5-DH in terms of both evaluation metrics (*mAP* and *F1 score*).

Table 12. Results of Faster R-CNN on CCTSDB2021 dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	61.9	48.7	45.7	46.0	61.1
F1 score	0.65	0.62	0.59	0.61	0.65

Table 13. Results of YOLOv4-TINY on CCTSDB2021 dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	62.6	64.2	53.7	62.3	64.7
F1 score	0.66	0.68	0.62	0.65	0.67

Table 14. Results of YOLOv5n on CCTSDB2021 dataset.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
***mAP* (%)**	68.7	72.5	54.8	60.7	71.3
F1 score	0.72	0.72	0.60	0.63	0.74

Table 15. Results of YOLOv5s on CCTSDB2021 dataset.

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**	**Experiment 5**
***mAP* (%)**	64.7	70.2	58.9	68.5	73.7
F1 score	0.70	0.72	0.63	0.69	0.76

Table 16. Results of YOLOv5-DH on CCTSDB2021 dataset.

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**	**Experiment 5**
***mAP* (%)**	71.9	67.6	62.2	65.2	73.8
F1 score	0.71	0.70	0.68	0.68	0.76

Table 17. Results of YOLOv5-TDHSA on CCTSDB2021 dataset.

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**	**Experiment 5**
***mAP* (%)**	69.1	73.4	62.1	69.7	74.7
F1 score	0.73	0.76	0.66	0.67	0.76

Table 18. Results of compared models on CCTSDB2021 dataset.

Model	***F1Score***	***mAP* (%)**	**Processing Speed (fps)**
Faster R-CNN	0.62	52.7	28
YOLOv4-Tiny	0.66	61.5	162
YOLOv5n	0.68	65.6	83
YOLOv5s	0.70	67.2	77
YOLOv5-DH	**0.71**	**68.1**	**70**
YOLOv5-TDHSA	**0.72**	**69.8**	**66**

6. Discussion

The incorporation of the proposed improvements into YOLOv5s resulted in overall better traffic sign detection. This was confirmed by a series of experiments conducted for evaluating and comparing the performance of the proposed models (YOLOv5-DH and YOLOv5-TDHSA) to that of YOLOv5s and three other state-of-the-art models, namely Faster R-CNN, YOLOv4-Tiny, and YOLOv5n, based on two datasets—TT100K and CCTSDB2021. The obtained results clearly demonstrate that both proposed models outperform all four models, in terms of the *mean average precision* (*mAP*) and *F1 score*.

Although both proposed models are better than the two-stage detection Faster R-CNN model, in terms of the model's size, number of parameters, and processing speed, they still have some shortcomings in this regard compared with the one-stage detection models (YOLOv4-Tiny, YOLOv5n, YOLOv5s). Therefore, in the future, some lightweight modules will be introduced into the proposed YOLOv5-TDHSA model (which is superior to the other proposed model YOLOv5-DH) in order to reduce its size and number of parameters, and increase its processing speed.

To check if the proposed models are significantly different statistically from the compared state-of-the-art models, we applied the (non-parametric) Friedman test [43,44] with the corresponding post-hoc Bonferroni–Dunn test [45,46], which are regularly used for the comparison of classifiers (more than two) over multiple datasets.

First, using the Friedman test, we measured the performances of the models, used in the experiments described in the previous section, across both datasets. Basically, the Friedman test shows whether the measured average ranks of models are significantly different from the mean rank expected, by checking the null hypothesis (stating that all models perform the same and the observed differences are merely random), based on the following formula:

$$\mathrm{T}_{x^2} = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right), \tag{7}$$

where k denotes the number of models, N denotes the number of datasets, and r_i. represents the average rank of the i-th model. In our case, $k = 6$ and $N = 2$.

Instead of Friedman's T_{x^2} statistic, we used the better Iman and Davenport statistic [47], which is distributed according to the F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom, as follows:

$$T_F = \frac{(N-1)T_{x^2}}{N(k-1) - T_{x^2}}\,. \tag{8}$$

Using (8), we calculated the following values: $T_F = 34$ for *F1 score* and $T_F = \infty$ for *mAP*. As both these values are greater than the critical values of 3.45 and 5.05 for six models and two datasets, with confidence levels of α = 0.10 and α = 0.05, respectively, we rejected the null hypothesis and concluded that there are significant differences between the compared models.

Next, we proceeded with a post-hoc Bonferroni–Dunn test, in which the models were compared only to a control model and not between themselves [44,48]. In our case, we used the proposed YOLOv5-TDHSA model as a control model. The advantage of the Bonferroni–Dunn test is that it is easier to visualize because it uses the same Critical Difference (CD) for all comparisons, which can be calculated as follows [48]:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}\,, \tag{9}$$

where q_α denotes the critical value for $\frac{\alpha}{k-1}$. When $k = 6$, $q_\alpha = 2.326$ for α = 0.10, and $q_\alpha = 2.576$ for α = 0.05 [48]. Then, the corresponding CD values, calculated according to (9), are equal to 4.352 and 4.819, respectively. Figure 9 shows the CD diagrams based on *F1 score* and *mAP*. As can be seen from Figure 9, the proposed YOLOv5-TDHSA model is significantly superior to Faster R-CNN on both evaluation metrics for both confidence levels, and achieves at least comparable performance to that of YOLOv4-Tiny on both evaluation metrics for both confidence levels, and to that of YOLOv5n on *F1 score* for both confidence levels. It is not surprising that the Bonferroni–Dunn test found YOLOv5-DH and YOLOv5-TDHSA similar to YOLOv5s, as both proposed models are based on it. Having incorporated only one YOLOv5s improvement into itself, naturally, YOLOv5-DH is reported by the Bonferroni–Dunn test as more similar to YOLOv5s than YOLOv5-TDHSA.

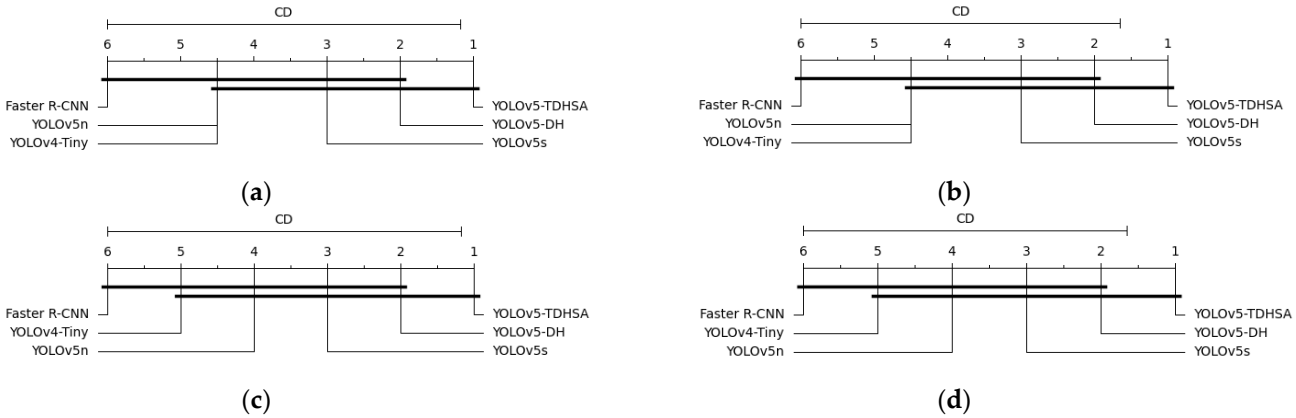

Figure 9. Critical difference (CD) comparison of YOLOv5-TDHSA (the control model) against other compared models with the Bonferroni–Dunn test, based on (**a**) *F1 score* with confidence level $\alpha = 0.05$, CD = 4.819; (**b**) *F1 score* with confidence level $\alpha = 0.10$, CD = 4.352; (**c**) *mAP* with confidence level $\alpha = 0.05$, CD = 4.819; (**d**) *mAP* with confidence level $\alpha = 0.10$, CD = 4.352 (any two models not connected by a thick black horizontal line are considered to have significant performance differences between each other).

7. Conclusions

We have proposed two novel models for accurate traffic sign detection, called YOLOv5-DH and YOLOv5-TDHSA, based on the YOLOv5s model with additional improvements. Firstly, a transformer self-attention module with stronger expression abilities was used in YOLOv5-TDHSA to replace the last layer of the 'Conv + Batch Normalization + SiLU' (CBS) structure in the YOLOv5s backbone. A similar module was added to the last layer of the YOLOv5-TDHSA's neck, so that the image information can be used more comprehensively. The features were mapped to the new three spaces for representation, thus improving the representation ability of the feature extraction. The multi-head mechanism used aims to realize the effect of multi-channel feature extraction. So, the transformer can increase the diversity of similarity computation between inputs and improve the ability of feature extraction. Secondly, a decoupled detection head was used in both proposed models to replace the YOLOv5s coupled head, which is responsible for the recognition and positioning on a feature map. As these two tasks have different focuses, resulting in a misalignment problem, the decoupled head uses two parallel branches—one responsible for the category recognition and the other responsible for positioning—which allows to improve the detection accuracy. However, as the decoupled head is not as fast as the coupled head, and due to the increase in the number of model parameters, the dimension was reduced through a 1×1 convolution before the decoupling to achieve balance between the speed and accuracy. Thirdly, for YOLOv5-TDHSA, a small-object detection layer was added to the YOLOv5s backbone and connected to the neck. At the same time, upsampling was used on the feature map of the neck to further expand the feature map. Supplemented by a group of adaptive anchor boxes, this new branch structure can not only ease the difficulty of small-object detection performed by YOLOv5-TDHSA, but can also compensate the feature losses caused by feature transmission with the increasing network depth.

Experiments conducted on two public datasets demonstrated that both proposed models outperform the original YOLOv5s model and three other state-of-the-art models (Faster R-CNN, YOLOv4-Tiny, YOLOv5n) in terms of the mean accuracy (*mAP*) and *F1 score*, achieving *mAP* values of 77.9% and 83.4% and *F1 score* values of 0.767 and 0.811 on the TT100K dataset, and *mAP* values of 68.1% and 69.8% and *F1 score* values of 0.71 and 0.72 on the CCTSDB2021 dataset, respectively, for YOLOv5-DH and YOLOv5-TDHSA. The results also confirm that the incorporation of the T and SA improvements into YOLOv5s leads to further enhancement, and a better performing model (YOLOv5-TDHSA), which

is superior to the other proposed model (YOLOv5-DH) that avails of only one YOLOv5s improvement (i.e., DH).

Author Contributions: Conceptualization, Z.J., W.B.; methodology, W.B.; validation, J.Z., C.D.; formal analysis, H.Z., L.Z.; writing—original draft preparation, W.B.; writing—review and editing, I.G.; supervision, I.G.; project administration, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This publication has emanated from research conducted with the financial support of the National Key Research and Development Program of China under grant no. 2017YFE0135700, the Tsinghua Precision Medicine Foundation under grant no. 2022TS003,and the MES by grant no. D01-168/28.07.2022 for NCDSC part of the Bulgarian National Roadmap on RIs.

Data Availability Statement: The data used in this study are openly available as per [38] and [40].

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

References

1. Saadna, Y.; Behloul, A. An overview of traffic sign detection and classification methods. *Int. J. Multimed. Inf. Retr.* **2017**, *6*, 193–210. [CrossRef]
2. Yıldız, G.; Dizdaroğlu, B. Traffic Sign Detection via Color and Shape-Based Approach. In Proceedings of 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 6–7 November 2019; pp. 1–5.
3. Shen, X.; Liu, J.; Zhao, H.; Liu, X.; Zhang, B. Research on Multi-Target Recognition Algorithm of Pipeline Magnetic Flux Leakage Signal Based on Improved Cascade RCNN. In Proceedings of 2021 3rd International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 8–11 November 2021; pp. 1–6.
4. Wang, X.; Wang, S.; Cao, J.; Wang, Y. Data-driven based tiny-YOLOv3 method for front vehicle detection inducing SPP-net. *IEEE Access* **2020**, *8*, 110227–110236. [CrossRef]
5. Rani, S.; Ghai, D.; Kumar, S. Object detection and recognition using contour based edge detection and fast R-CNN. *Multimed. Tools Appl.* **2022**, *81*, 42183–42207. [CrossRef]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pat. Analys. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Fan, J.; Huo, T.; Li, X. A Review of One-Stage Detection Algorithms in Autonomous Driving. In Proceedings of 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 210–214.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 779–788.
9. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 7263–7271.
10. Lv, N.; Xiao, J.; Qiao, Y. Object Detection Algorithm for Surface Defects Based on a Novel YOLOv3 Model. *Processes* **2022**, *10*, 701. [CrossRef]
11. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
12. Ma, W.; Zhou, T.; Qin, J.; Zhou, Q.; Cai, Z. Joint-attention feature fusion network and dual-adaptive NMS for object detection. *Knowl. Based Syst.* **2022**, *241*, 108213. [CrossRef]
13. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
14. Hu, J.; Shen, L.; Sun, G. Squeeze-And-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 7132–7141.
15. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 3–19.
16. Liang, H.; Zhou, H.; Zhang, Q.; Wu, T. Object Detection Algorithm Based on Context Information and Self-Attention Mechanism. *Symmetry* **2022**, *14*, 904. [CrossRef]
17. Lou, Y.; Ye, X.; Li, M.; Li, H.; Chen, X.; Yang, X.; Liu, X. Object Detection Model of Cucumber Leaf Disease Based on Improved FPN. In Proceedings of the 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Beijing, China, 3–5 October 2022; pp. 743–750.
18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. *arXiv* **2018**, arXiv:1803.01534.
20. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]
21. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055. [CrossRef]

22. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [CrossRef]
23. Dixit, U.D.; Shirdhonkar, M.; Sinha, G. Automatic logo detection from document image using HOG features. *Multimed. Tools Appl.* **2022**, *82*, 863–878. [CrossRef]
24. Kim, J.; Lee, K.; Lee, D.; Jhin, S.Y.; Park, N. DPM: A novel training method for physics-informed neural networks in extrapolation. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 8146–8154. [CrossRef]
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
26. Niepceron, B.; Grassia, F.; Moh, A.N.S. Brain Tumor Detection Using Selective Search and Pulse-Coupled Neural Network Feature Extraction. *Comput. Inform.* **2022**, *41*, 253–270. [CrossRef]
27. Du, L.; Zhang, R.; Wang, X. Overview of Two-Stage Object Detection Algorithms. *Proc. J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [CrossRef]
28. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
29. Bouabid, S.; Delaitre, V. Mixup regularization for region proposal based object detectors. *arXiv* **2020**, arXiv:2003.02065.
30. Chen, Y.; Wang, J.; Dong, Z.; Yang, Y.; Luo, Q.; Gao, M. An Attention based YOLOv5 Network for Small Traffic Sign Recognition. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, 1–3 June 2022; pp. 1158–1164.
31. Liu, X.; Jiang, X.; Hu, H.; Ding, R.; Li, H.; Da, C. Traffic Sign Recognition Algorithm Based on Improved YOLOv5s. In Proceedings of the 2021 International Conference on Control, Automation and Information Sciences (ICCAIS), Xi'an, China, 14–17 October 2021; pp. 980–985.
32. Chen, X. Traffic Lights Detection Method Based on the Improved YOLOv5 Network. In Proceedings of the 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, 12–14 October 2022; pp. 1111–1114.
33. Chen, Z.; Guo, H.; Yang, J.; Jiao, H.; Feng, Z.; Chen, L.; Gao, T. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement* **2022**, *201*, 111655. [CrossRef]
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
35. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
36. Ballantyne, G.H. Review of sigmoid volvulus. *Dis. Colon Rectum* **1982**, *25*, 823–830. [CrossRef] [PubMed]
37. Wang, J.; Chen, Y.; Gao, M.; Dong, Z. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *arXiv* **2021**, arXiv:2112.08782. [CrossRef]
38. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 2110–2118.
39. Chen, J.; Jia, K.; Chen, W.; Lv, Z.; Zhang, R. A real-time and high-precision method for small traffic-signs recognition. *Neural Comput. Appl.* **2022**, *34*, 2233–2245. [CrossRef]
40. Zhang, J.; Zou, X.; Kuang, L.-D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A more comprehensive traffic sign detection benchmark. *Hum. Cent. Comput. Inf. Sci.* **2022**, *12*, 23.
41. Zhang, J.; Huang, M.; Jin, X.; Li, X. A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms* **2017**, *10*, 127. [CrossRef]
42. Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Sangaiah, A.K. Spatial and semantic convolutional features for robust visual object tracking. *Multimed. Tools Appl.* **2020**, *79*, 15095–15115. [CrossRef]
43. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
44. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [CrossRef]
45. Dunn, O.J. Multiple Comparisons among Means. *J. Am. Stat. Assoc.* **1961**, *56*, 52–64. [CrossRef]
46. Lin, Y.; Hu, Q.; Liu, J.; Li, J.; Wu, X. Streaming feature selection for multilabel learning based on fuzzy mutual information. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1491–1507. [CrossRef]
47. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the fbietkan statistic. *Commun. Stat. Theory Methods* **1980**, *9*, 571–595. [CrossRef]
48. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.