Small Object Detection in Intelligent Transportation Systems: Design and Optimization of the TSTD Model

Tao Zhang[1,*], Yihe Jin[1], Jialin Wang[1]

[1]Tianjin University of Technology and Education, Tianjin, China
[*]Corresponding author

Abstract: *With the rapid development of transportation systems, road safety and traffic management have become crucial. Efficient traffic sign detection and recognition enhance traffic flow and safety. This paper proposes a Traffic Sign Tiny Detector (TSTD) algorithm to improve the performance of existing small object detection models. The TSTD algorithm utilizes efficientFormerv2, specifically designed for small objects, and optimizes the loss function with a normalized Wasserstein distance loss. It also employs the C2f_DBB module to replace traditional downsampling, preventing excessive loss of small object information. EfficientFormerv2 offers higher efficiency and lower computational cost, significantly reducing the model's complexity and training time while maintaining high accuracy. The C2f_DBB module, with its improved feature fusion and dual-branch structure, enhances the model's ability to detect small objects, ensuring high-precision recognition of tiny traffic signs. Extensive comparative experiments verify the model's advantages in traffic sign detection. Results show that TSTD significantly improves key performance metrics, such as mean Average Precision (mAP), over baseline models. In summary, the proposed TSTD can more accurately detect traffic signs, contributing to advancements in intelligent traffic management and improving road safety and traffic efficiency.*

Keywords: *Traffic Sign; efficientFormerv2; C2f_DBB; Normalized Wasserstein Distance*

1. Introduction

Accurate traffic sign recognition is crucial for road safety in modern intelligent transportation systems. As urbanization accelerates and traffic networks become more complex, timely and accurate road information for drivers is essential. Small target traffic signs, such as those at a distance or partially obscured, play a key role in maintaining smooth traffic flow and preventing accidents. However, detecting these signs under various weather and lighting conditions presents significant challenges, especially when located at the road edge or in complex backgrounds.

Traditional traffic sign detection techniques rely on manual feature extraction and simple image processing, which perform poorly with small target traffic signs and in complex environments. Early systems may only be effective under specific lighting conditions or may struggle to distinguish signs from cluttered backgrounds. With the development of deep learning technology, improving the accuracy and efficiency of small object detection using advanced computer vision algorithms has become a research hotspot. These algorithms can automatically learn and extract complex features from images, greatly improving detection accuracy and reliability.

However, despite their excellent performance in laboratory conditions, challenges remain in practical applications. These advanced algorithms often require significant computational resources, which may be difficult to meet in scenarios requiring fast real-time processing, particularly with video streams or large-scale real-time traffic monitoring data. Additionally, the high real-time requirements of intelligent transportation systems mean that any delay can affect driving decisions, increasing the risk of accidents.

Current mainstream small object detection methods primarily include convolutional neural network (CNN)-based models such as SSD (Single Shot MultiBox Detector)[1], YOLO (You Only Look Once)[2], and Faster R-CNN (Faster Regions with Convolutional Neural Network)[3]. These models can achieve high detection accuracy in many cases, but they often encounter difficulties when dealing with extremely small or complex background targets, especially when the target size is very small or the target-to-background contrast is low. Furthermore, these methods require high computational costs and storage resources, limiting their application on resource-constrained devices such as mobile devices or edge computing devices. Therefore, developing high-precision traffic sign detection systems that can run efficiently in resource-limited environments is essential to meet the high real-time and reliability requirements of intelligent transportation systems.

To overcome these limitations, this study proposes a novel deep learning model, the TrafficSignTinyDetector (TSTD), as shown in Figure 1. The main contributions of this work are:

1) The model introduces efficientFormerv2[4] as the backbone network, improving the efficiency and accuracy of feature extraction, enabling the model to quickly and accurately recognize traffic signs in complex environments.

2) The C2f_DBB[5] module is used to replace the traditional C2f module, enhancing the model's ability to detect small targets and improving sensitivity and accuracy in small target detection.

3) The introduction of the Normalized Wasserstein Distance (NWD)[6] helps the model more finely handle minor variations in target distribution, enhancing detection accuracy and robustness.

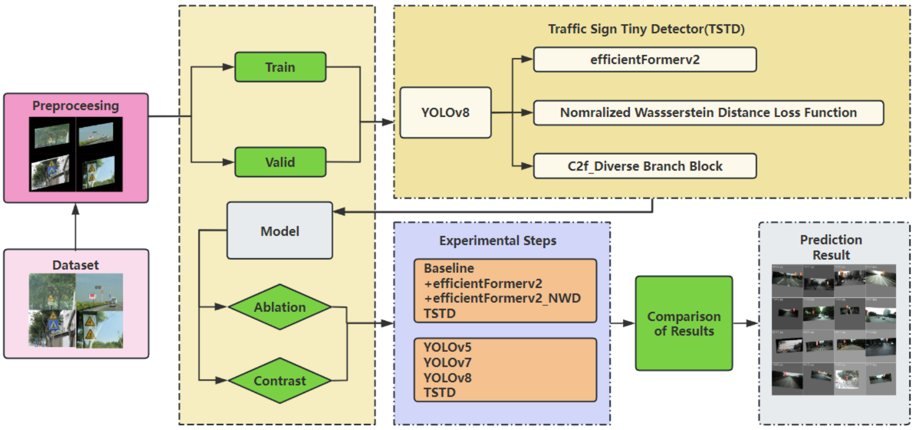

Figure 1: Model Workflow.

2. TrafficSignTinyDetector (TSTD) Architecture Design

The Traffic Sign Tiny Detector (TSTD) system is based on the improved YOLOv8 architecture, using efficientFormerv2 as the backbone network to enhance feature extraction efficiency and accuracy. This enables better handling of complex environment images, especially for small or distant traffic signs.

The detection head of the TSTD system uses the novel C2f_DBB module to replace the traditional C2f module, designed to enhance the response capability to small targets. By optimizing feature fusion and processing mechanisms, the C2f_DBB module can more accurately locate and recognize small traffic signs, improving overall detection accuracy.

Additionally, the TSTD system introduces the Normalized Wasserstein Distance (NWD) into the loss function, focusing on the differences between actual and expected outputs during training. This method improves the model's generalization ability for different sizes and types of traffic signs and significantly enhances robustness in complex backgrounds.

Overall, the TSTD system forms an efficient, accurate, and adaptable traffic sign detection system. It is suitable for typical urban and suburban road environments and can work effectively on highways and under adverse weather conditions, providing robust support for real-time applications in intelligent transportation systems.

3. Related Work

3.1. Data Augmentation

Large datasets are required in object detection for high precision and robustness. Given the small scale and limited diversity of traffic sign datasets, data augmentation is crucial. This study uses mosaic and mixup data augmentation methods to enhance data diversity during training for the TrafficSignTinyDetector (TSTD) model.

The mosaic data augmentation algorithm stitches multiple images together proportionally to form a new training image, enabling the model to recognize targets in smaller regions. This method is derived from the CutMix data augmentation algorithm, with the main difference being that CutMix typically uses two images for stitching, whereas Mosaic uses four. This design helps the model improve recognition ability for small or partially obscured traffic signs when handling real-world scenarios.

Additionally, the mixup algorithm blends two images at the pixel level to generate new training samples. This method not only increases the diversity of training data but also improves the model's robustness to noise in the images. Mosaic augmentation is typically turned off in the last 10 training cycles to refine the model's fit to real-world scene data.

By applying these data augmentation techniques, the TSTD model can effectively improve the detection performance of small traffic signs in complex environments, enhancing the model's generalization ability and robustness.

3.2. Object Detectors

Two-stage object detectors divide the detection process into two stages: the first stage extracts regions where objects are located, and the second stage uses CNN to classify the regions. These detectors usually have higher precision, but the multiple steps make them slower, unsuitable for real-time detection.

RCNN (Region-CNN) selects a set of object candidate boxes through selective search, resizes them to a fixed size, and uses a CNN model to extract features, followed by SVM (Support Vector Machine) for prediction and target classification. SPPNet adds a pyramid pooling layer after the last convolutional layer. Faster R-CNN uses Region Proposal Network (RPN) instead of selective search to generate proposal windows and shares CNN features. FPN (Feature Pyramid Network) constructs different scales of images or feature maps for model training and testing, enhancing robustness to different sizes of targets.

Single-stage object detectors output detection results in a single training step, significantly improving detection speed but usually having lower precision than two-stage detectors due to simplified steps. The TSTD algorithm gradually balances speed and precision during its development. SSD (Single Shot Multibox Detector)[1] is based on a feed-forward convolutional network, generating fixed-size detection boxes, scoring object instances within them, and using non-maximum suppression to produce the final result. YOLO (You Only Look Once) takes the entire image as input, directly regressing detection box locations and their respective categories in the output. Swin Transformer[7] improves upon ViT[8] by performing self-attention mechanism calculations through window scaling, introducing locally aggregated information.

3.3. Application of efficientFormerv2

EfficientFormerv2 is a highly efficient Transformer network architecture specifically designed for handling large-scale datasets and complex image features in computer vision tasks. In this study's Traffic Sign Tiny Detector (TSTD) model, efficientFormerv2 plays a crucial role as the backbone network. Its self-attention mechanism can effectively capture long-distance dependencies in images, which is particularly important for recognizing and distinguishing small-sized traffic signs in complex backgrounds. Compared to traditional convolutional neural networks, efficientFormerv2 provides a wider field of view and finer feature representation, enhancing the model's ability to recognize traffic sign details.

Additionally, efficientFormerv2 optimizes resource allocation during computation, reducing the model's operational computational cost. This allows efficientFormerv2 to perform well not only in high-performance computing environments but also on resource-constrained mobile devices and edge computing platforms, which is critical for real-time traffic sign detection systems. EfficientFormerv2 can improve the model's performance in various complex traffic scenarios, especially when detecting small targets and traffic signs in dynamic environments. With this efficient network structure, the model can achieve fast and accurate traffic sign recognition, supporting rapid decision-making and response in intelligent transportation systems.

3.4. YOLOv8 Algorithm

Since the open-source release of YOLOv8 code, extensive testing has shown an unprecedented balance between precision and speed. YOLOv8 optimizes and improves upon YOLOv5, with key enhancements including replacing the C3 module with the C2f module for lightweight processing. The C3 module combines CSPNet's shunt concept with a residual structure, while the C2f module adds more gradient flow branches in parallel, gathering more gradient flow information through the ELAN (Effective Long-Range Aggregation Network) module, enhancing precision while maintaining lightweight processing. The Neck stage removes the convolution layer before upsampling, reducing the algorithm's size and improving performance.

The Head part is changed to a Decoupled-Head, which uses two convolutions separately for classification and regression. The regression head's channel number is modified to 4×reg_max due to the use of the DFL concept. YOLOv8 also adopts the Anchor Free method, transitioning to an Anchor Free era. The anchor Free method represents objects through multiple key points or central points and boundary information, making it more suitable for small object detection, especially in complex backgrounds.

4. TrafficSignTinyDetector Model Construction

4.1. Overall Network Structure

YOLOv8 is an improved version of the YOLOv5 benchmark model, offering not only better precision but also enhanced performance and lightweight design. The algorithm replaces the C3 module with the C2f module, using more gradient flow branches in parallel to gather richer gradient information. Gradient flow branches have proven effective in practice for improving algorithm precision. The model also changes the SPP (Spatial Pyramid Pooling) structure to the SPPF (Spatial Pyramid Pooling-Fast) structure, achieving similar effects while reducing execution time by half. The prediction head is also changed to DecoupledHead, accelerating algorithm convergence and improving performance in end-to-end predictions.

The official release includes five configurations: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. However, the original model still faces challenges in detecting small traffic signs, particularly in dense or tiny sign locations, lacking contextual information, and suffering from the negative effects of discontinuities caused by negative samples. This study improves upon the YOLOv8n algorithm and strengthens the backbone for enhanced environmental understanding and the improved detection head algorithm for cases with many small target signs. The baseline YOLOv8 network structure is shown in Figure 2, and the improved TSTD (TrafficSignTinyDetector) network structure is shown in Figure 3.

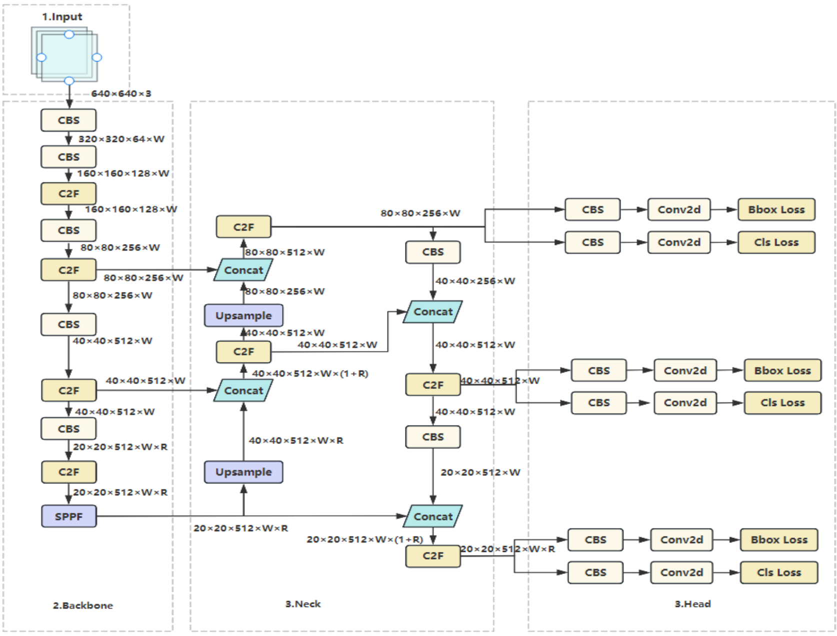

Figure 2: YOLOv8 Network Structure.

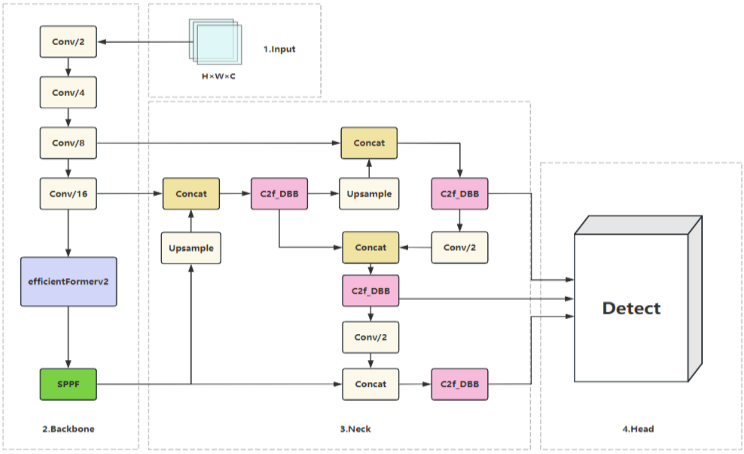

Figure 3: TSTD Network Structure.

4.2. Overall Network Structure

In the field of object detection, the attention mechanism is considered an effective method for capturing long-distance relationships between targets. To capture global features and complex contextual information in images, Alexey et al. proposed Vision Transformer (ViT), which slices and unfolds images to capture relationships within them. However, ViT and its variants still have higher latency or more parameters than lightweight CNNs, even compared to MobileNet[9] from years ago. In practice, latency and size are crucial for efficient deployment on resource-constrained hardware. To address this issue, Yanyu Li et al. reexamined ViT's design choices and proposed an improved super-network, efficientFormer[10], with low latency and high parameter efficiency, and further introduced a fine-grained joint search strategy to propose efficientFormerv2[4], a strategy that finds effective architectures by simultaneously optimizing latency and parameter count.

To enhance YOLOv8's performance in small traffic sign detection, this study introduces efficientFormerv2 as the backbone network in the Traffic Sign Tiny Detector (TSTD) model. EfficientFormerv2 is a highly efficient Transformer network architecture designed for processing large-scale datasets and complex image features. Its introduction brings several key improvements to the model:

4.2.1. Attention Mechanism

EfficientFormerv2 uses a self-attention mechanism[11] that can effectively capture long-distance dependencies in images. This is particularly important for recognizing and distinguishing small-sized traffic signs in complex backgrounds. Compared to traditional convolutional neural networks, efficientFormerv2 provides a wider field of view and finer feature representation, allowing the model to more accurately recognize traffic sign details.

Furthermore, the traditional ViT and EfficientFormer attention modules use the MHSA+FFN structure. To improve the performance of the attention module without increasing the model's size and latency, efficientFormerv2 adopts two methods to improve MHSA, described by equations (1) and (2) for traditional and multi-head attention mechanisms, respectively. Figure 4 depicts the improvements in MHSA in v2. Firstly, by adding 3×3 depth-wise convolution to the value matrix (V) in the depth direction, local information is injected. Secondly, adding fully connected layers (talking head) in the head dimension enables communication between attention heads. These improvements enhance the performance of the attention mechanism while maintaining low computational overhead.

4.2.2. Computational Efficiency and Precision

EfficientFormerv2 optimizes resource allocation during computation, significantly reducing the model's operational computational cost. Despite its reduced computational complexity, it maintains high precision in feature extraction. This allows efficientFormerv2 to perform well not only in high-performance computing environments but also on resource-constrained mobile devices and edge computing platforms, supporting real-time traffic sign detection systems.

4.2.3. Optimized Feature Fusion

EfficientFormerv2 optimizes feature fusion, enabling more effective integration of feature information at different levels. This optimization improves the model's performance in handling multiscale targets, making it more effective in detecting distant or partially obscured traffic signs.

4.3. C2F_DBB

In the Traffic Sign Tiny Detector (TSTD) model, to further enhance the detection performance of small traffic signs, this study introduces the C2F_DBB (C2f with Diverse Branch Block) module based on YOLOv8. The C2F_DBB module improves feature fusion and processing mechanisms, enhancing the response capability and detection accuracy for small targets.

4.3.1. Feature Fusion and Lightweight Design

The C2F_DBB module enhances feature fusion capability through a multi-branch design. Multiple branches capture and fuse features from different perspectives, improving the model's ability to detect targets in small and complex backgrounds, thus improving overall detection performance.

Despite introducing multiple branches and complex feature fusion mechanisms, the C2F_DBB module maintains a lightweight design through efficient convolution operations and optimized network structure. This ensures excellent performance in high-performance computing environments while also being suitable for deployment on resource-constrained mobile devices and edge computing platforms, providing technical support for real-time traffic sign detection.

4.3.2. Efficient Convolution Operations

The C2F_DBB module adopts a combination of 3×3 depth-wise convolution (DWCONV) and other convolution operations, improving feature extraction efficiency and precision. Depth-wise convolution effectively captures local features, while other convolution operations enhance the global representation of features. The combination of both maintains computational efficiency while improving detection precision.

4.4. NWD Loss Function

In traditional YOLOv8, the coordinate loss function is based on Intersection over Union (IoU)[2] calculation and further optimized as Complete Intersection over Union (CIoU)[11]. Class loss uses cross-entropy loss, and object loss uses binary cross-entropy loss. The weighted sum of these three constitutes the final loss function. However, in the detection of tiny traffic signs, even very small positional deviations can cause significant changes in IoU value because IoU metrics remain constant across different scales.

For example, in an image, tiny traffic signs may occupy only a few pixels, making IoU-based metrics susceptible to noise. This noise can cause excessive sensitivity to minor positional changes in small targets. To address this issue, this study introduces Gaussian distribution to optimize the loss function for small object detection scenarios.

By comparing the probability distribution differences between predicted and actual bounding boxes, this study can measure their similarity more accurately, rather than directly computing positional differences. Using Gaussian distribution to describe and compare these distributions helps avoid unstable metrics due to minimal positional deviations. The probability density function of a two-dimensional Gaussian distribution is shown in equation (1):

$$f(x) = \frac{1}{2\pi |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \tag{1}$$

Where x is the position vector containing coordinates (x, y), μ is the mean composed of the coordinate values (x, y), and Σ is the positive definite covariance matrix with corresponding variance values on the diagonal.

5. Experiments and Results Analysis

5.1. Experimental Platform

The experiments in this paper were conducted on an Ubuntu 20.04 system with an NVIDIA GeForce RTX 4090 GPU, 24 GB of VRAM, and an Intel(R) Xeon(R) Platinum 8352V CPU. The deep learning framework used was Pytorch 1.11.0, with Python 3.8.10 as the programming language. CUDA 11.3 and cuDNN 11.3.109 were installed to support GPU acceleration.

5.2. The CCTSDB Dataset

The dataset used in this study is CCTSDB2021 (Chinese Common Traffic Sign Database 2021). A total of 2000 images were selected, covering different lighting conditions (e.g., cloudy, foggy, nighttime, rainy, snowy, and sunny weather) and traffic signs. Some images were augmented by changing horizontal dimensions, adding salt-and-pepper noise, and adjusting brightness. The dataset was divided into training, validation, and test sets: 1200 images were used as the training set, with at least 400 targets per category; the remaining 800 images were evenly split into validation and test sets, ensuring an even distribution of targets per category. Since the latter part of the CCTSDB contains images extracted from driving videos with high scene repetition, fewer images were selected to avoid excessive similarity in the dataset.

CCTSDB2021 is a large-scale image dataset targeting Chinese traffic signs, launched in 2021. It covers the diversity and complexity of Chinese traffic signs, including different sizes, shapes, and partial occlusions. The dataset contains up to 10,000 high-resolution images, encompassing various categories such as warning signs, prohibition signs, and instruction signs. Each image comes with detailed annotations, including the sign's category, location, and size, provided in a standard annotation format for easy use and reference by algorithm developers. CCTSDB2021 also features a standard evaluation system for fair assessment of participating models' recognition and detection performance.

5.3. Experimental Dataset and Setup

The experimental setup largely follows the optimized parameter settings of YOLOv8. Mosaic and mixup were used during data preprocessing, with mosaic turned off in the last 10 epochs. Degrees were set to 10 degrees, image scaling ratio to 0.1, adaptive anchor box calculation was used, and input image size was 640×640. Random gradient descent strategy (SGD) was used to optimize network parameters during training, with a learning rate set to 0.01, learning rate momentum to 0.937, weight decay to 0.0005, batch size to 16, and epoch to 300. NWD loss function IOU_Ratio was set to 0.5.

5.4. Evaluation Metrics

In the model evaluation phase, this study used precision, recall, F1-score, and mean Average Precision (mAP) as the main evaluation metrics. The detailed description of each metric is as follows:

Precision mainly evaluates the model's ability to accurately recognize positive samples, calculated as shown in equation (2); recall reflects the model's performance in capturing positive samples, calculated as shown in equation (3); F1-score, as the harmonic mean of precision and recall, provides an overall balanced assessment of the model between these two metrics, calculated as shown in equation (4); mAP is a comprehensive metric that considers the accuracy of the model at various recall levels and averages these accuracies, making it an important practical metric, calculated as shown in equation (5).

$$\mathrm{P} = TP/(TP+FP)\times 100\% \tag{2}$$

$$R = TP/（TP+FN）\times 100\% \tag{3}$$

$$F_1 Score = (2\times P\times R)/(P+R) \tag{4}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP(i)\times 100\% \tag{5}$$

Where TP (true positive) is the number of correctly predicted positive cases, FP (false positive) is the number of incorrectly predicted positive cases, and FN (false negative) is the number of incorrectly predicted negative cases.

5.5. Experimental Results Analysis

5.5.1. Ablation Experiment Results

During the ablation experiments, this study used the baseline model YOLOv8 and gradually added model components for comparison to examine the impact of each component on model performance. These experiments validated the effectiveness of the new components and their contribution to model improvement.

During the experiments, the mAP of each model was compared to evaluate performance under different configurations. Figure 4 shows the mAP fitting curves of each model under different settings, illustrating the convergence of the model during training. Table 1 summarizes the experimental results under each configuration, detailing the specific impact of component combinations on model performance.

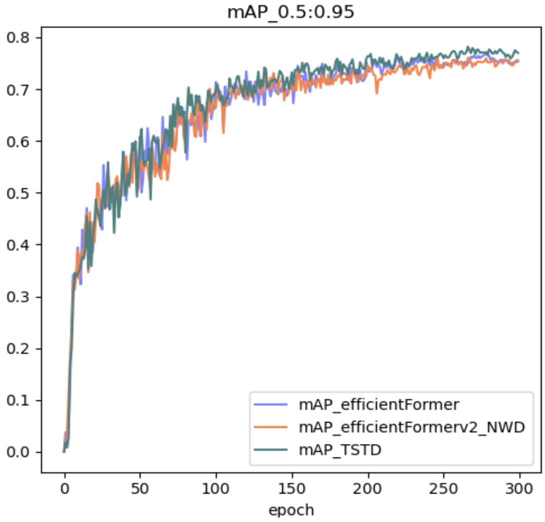

Figure 4: Ablation Experiment Fitting Curves.

Table 1: Ablation Experiment Results.

Algorithm Model	mAP@.5:0.95/%	Precision	Recall
efficientFormerv2	0.768	0.894	0.904
efficientFormerv2 NWD	0.761	0.898	0.903
TSTD	0.782	0.895	0.921

5.5.2. Comparative Experiment Results

This study proposes a new small target traffic sign detection model, TSTD, and demonstrates its superior performance through a series of ablation experiments. During the validation of its performance, TSTD was compared with the traditional YOLOv8 network architecture, showing that TSTD outperforms YOLOv8 in terms of performance.

To further validate the accuracy of TSTD, other architectures, including YOLOv5 and YOLOv7, were also selected for comparison. In the comparative results, this study evaluated the performance of each model in terms of mAP@.5:0.95/%, precision, and recall. The results show that TSTD significantly outperforms other models across all evaluation metrics.

Figure 5 clearly shows the performance differences between TSTD and other models. Table 2 details the comparative results between TSTD and other models, further proving the significant advantages of TSTD in small target traffic sign detection[12-15].

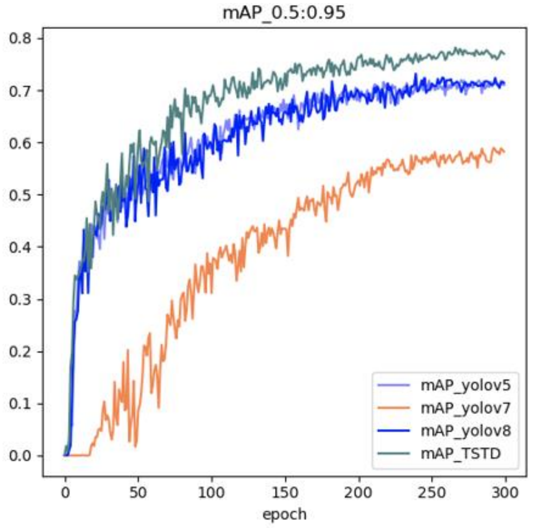

Figure 5: Comparative Experiment Fitting Curves.

Table 2: Comparative Experiment Results.

Algorithm Model	mAP@.5:0.95/%	Precision	Recall
YOLOv5	0.719	0.882	0.898
YOLOv7	0.589	0.886	0.787
YOLOv8n	0.731	0.879	0.914
TSTD	0.782	0.895	0.921

On the other hand, in the application scenarios of tiny traffic sign detection, this study expects the model to accurately recognize every traffic sign (maintaining high recall) while accurately predicting the signs (maintaining high precision). Using only precision or recall as performance metrics may lead to biased results. If only precision is emphasized, the model may become overly cautious, detecting signs only when it is very certain. Conversely, if only recall is emphasized, the model may become overly aggressive, making incorrect predictions in areas where there are no signs.

To address this issue, this study introduces the F1-score as an evaluation metric because it considers both precision and recall, providing a more comprehensive evaluation of model performance. The F1-score is the harmonic mean of precision and recall, balancing the two metrics, enabling the model to handle various scenarios in practical applications without being overly cautious or aggressive. Figure 6 shows the F1-scores of different models.

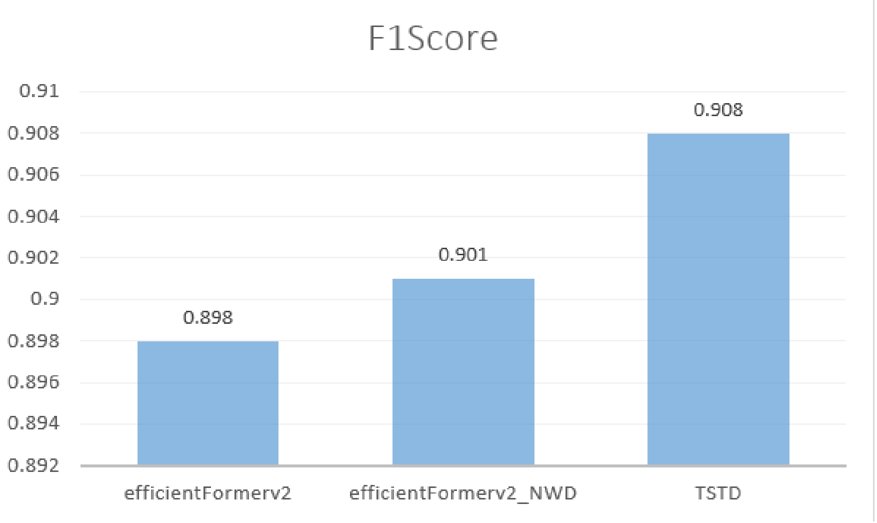

Figure 6: F1-Score.

6. Conclusions

This paper proposes the TSTD model for small target traffic sign detection, validated through extensive experiments. Compared to YOLOv8, YOLOv5, and YOLOv7, TSTD shows significant advantages in mAP, precision, and recall. By incorporating efficientFormerv2 as the backbone, the improved C2F_DBB module, and the NWD loss function, TSTD excels in complex background and small target detection. Additionally, the introduction of the F1-score provides a comprehensive evaluation of model performance, ensuring the model is neither overly cautious nor overly aggressive in practical applications. Comprehensive experimental results indicate that TSTD has significant potential and practical application value in real-time applications and traffic sign detection in intelligent transportation systems.

References

[1] Liu, Wei, Anguelov, Dragomir, Erhan, Dumitru, Szegedy, Christian, Reed, Scott, Fu, Cheng-Yang, and Berg, Alexander C. SSD: Single Shot MultiBox Detector. arXiv, (2015).
[2] Redmon, Joseph, Divvala, Santosh, Girshick, Ross, and Farhadi, Ali. You Only Look Once: Unified, Real-Time Object Detection. arXiv, (2015).
[3] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2017).
[4] Li, Yanyu, Hu, Ju, Wen, Yang, Evangelidis, Georgios, Salahi, Kamyar, Wang, Yanzhi, Tulyakov, Sergey, and Ren, Jian. Rethinking Vision Transformers for MobileNet Size and Speed. arXiv, (2023).
[5] Ding, Xiaohan, Zhang, Xiangyu, Han, Jungong, and Ding, Guiguang. Diverse Branch Block: Building a Convolution as an Inception-like Unit. arXiv, (2021).
[6] Wang, Jinwang, Xu, Chang, Yang, Wen, and Yu, Lei. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. arXiv, (2021).
[7] Liu, Ze, Lin, Yutong, Cao, Yue, Hu, Han, Wei, Yixuan, Zhang, Zheng, Lin, Stephen, and Guo, Baining. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv, (2021).
[8] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, Uszkoreit, Jakob, and Houlsby, Neil. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv, (2020).
[9] Howard, Andrew G., Zhu, Menglong, Chen, Bo, Kalenichenko, Dmitry, Wang, Weijun, Weyand, Tobias, Andreetto, Marco, and Adam, Hartwig. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, (2017).
[10] Li, Yanyu, Yuan, Geng, Wen, Yang, Hu, Ju, Evangelidis, Georgios, Tulyakov, Sergey, Wang, Yanzhi, and Ren, Jian. EfficientFormer: Vision Transformers at MobileNet Speed. arXiv, (2022).
[11] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Polosukhin, Illia. Attention Is All You Need. arXiv, (2017).
[12] Zheng, Zhaohui, Wang, Ping, Liu, Wei, Li, Jinze, Ye, Rongguang, and Ren, Dongwei. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. arXiv, (2019).
[13] Hu, Jie, Shen, Li, and Sun, Gang. Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2019).
[14] Zhang, Jianming, Zou, Xin, Kuang, Lide, and others. CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark. Human-centric Computing and Information Sciences, (2022).
[15] Zhang, Jianming, Wang, Wei, Lu, Chaoqiang, and others. Lightweight Deep Network for Traffic Sign Classification. Annals of Telecommunications, (2020).