Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

TSD-Net: A network for real-time detection of traffic signs by high-resolution dominant feature fusion

Shuo Hu

hus@ysu.edu.cn

Yanshan University

Tianqi Liu

Yanshan University

Liyang Han

Yanshan University

Research Article

Keywords: Deep convolutional neural networks, Traffic sign detection, Feature fusion, Dynamic convolution

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11678123/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

TSD-Net: A network for real-time detection of traffic signs by high-resolution dominant feature fusion

Shuo Hu[1,2*], Tianqi Liu[1] and Liyang Han[3]

[1*]School of Electrical Engineering, Yanshan University, Qinhuangdao, 066004, Hebei, China.
[2]Hebei Key Laboratory of Intelligent Rehabilitation and Neuroregulation, Yanshan University, Qinhuangdao, 066004, Hebei, China.
[3]Suzhou Inovance Technology Co., Ltd., Suzhou, Jiangsu, China.

*Corresponding author(s). E-mail(s): hus@ysu.edu.cn;
Contributing authors: liutianqi@stumail.ysu.edu.cn; hanliyang@inovance.com;

Abstract

Traffic sign detection in complex road environments requires high accuracy under strict real-time constraints. Small object size, occlusion, illumination changes, and adverse weather conditions degrade feature quality, whereas high-capacity detectors are difficult to deploy on resource-constrained edge devices. To meet these requirements, we develop TSD-Net, a lightweight traffic sign detector designed for real-time operation and based on high-resolution dominant feature fusion. First, the backbone uses deformable dynamic convolution to extract discriminative features while suppressing redundant responses. Second, the High-Resolution Dominant Feature Pyramid Network (HRDFPN) preserves fine spatial information and improves small-object representation through unidirectional feature fusion. Third, the Semantic Information Fusion Head (SIFH) decouples classification and localization while integrating semantic and boundary information with low computational overhead. Experiments on Tsinghua–Tencent 100K and the Chinese Traffic Sign Detection Benchmark 2021 show that TSD-Net balances detection accuracy, model size, and inference speed. Deployment on the NVIDIA Jetson Orin NX further demonstrates its feasibility for real-time applications on edge devices.

Keywords: Deep convolutional neural networks, Traffic sign detection, Feature fusion, Dynamic convolution

1 Introduction

Traffic sign detection (TSD) supports advanced driver assistance systems and autonomous driving [1, 2]. Early methods used handcrafted cues such as sliding windows, edges, and color transformations [3–5]. Their reliance on manually designed features makes them sensitive to lighting and weather. Deep detectors [6–8], including two-stage Faster R-CNN [9] and one-stage YOLO [10], offer different accuracy–speed trade-offs, but distant signs remain difficult to localize.

A traffic sign may occupy only 30×30 pixels in a 2048×2048 image. Downsampling can suppress the resulting weak features, while class imbalance, occlusion, and changing illumination further complicate detection (Fig. 1). Increasing model complexity can improve accuracy [11], but resource constraints limit its use on edge devices. Receptive-field optimization [12] and adaptive multi-scale processing [13] are alternative approaches; the central question here is how to preserve fine details while controlling model size and inference cost.

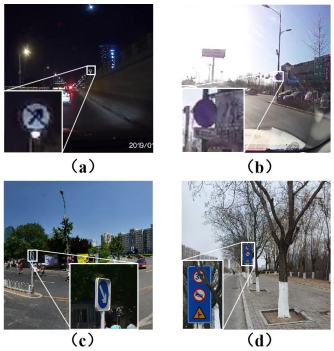

Fig. 1 Traffic sign imaging conditions: (a) nighttime, (b) bright light, (c) tilted sign, and (d) occlusion

We propose TSD-Net, which combines the DYPNet backbone, a high-resolution dominant feature pyramid network (HRDFPN), and a semantic information fusion head (SIFH). Its main contributions are:

1. A deformable dynamic kernel with skip connections (D2KSC) combines adaptive sampling and input-dependent weights while retaining shallow features.
2. A unidirectional HRDFPN preserves high-resolution features, and SIFH fuses semantic and spatial information separately for classification and localization.
3. Adaptive threshold focal loss (ATFL) addresses sample imbalance. Detection experiments, module ablations, and an edge-device benchmark evaluate the resulting accuracy–efficiency trade-off.

2 Related Work

2.1 Small-object detection

Small objects have limited spatial support and weak visual cues; the COCO small-object category uses an area below 32^2 pixels. Multi-scale feature representations improve fine-grained detection [14, 15]. Attention-enhanced Faster R-CNN [16] and perceptual feature enhancement [17] address weak small-object responses, while DSF-YOLO targets traffic signs in adverse weather [18]. Additional enhancement and fusion can increase training and inference cost.

2.2 Feature pyramid networks

FPN combines features at several resolutions [19]. BiFPN uses weighted cross-scale connections [20], and ASFF adapts fusion spatially [21]. For semantic segmentation, MCANet uses multi-scale convolutional attention and aggregation [22], while AEDFPN refines features through aggregation, enhancement, and top-down distribution [23]. HRDFPN instead emphasizes high-resolution detail retention for traffic sign detection.

3 Method

Figure 2 summarizes TSD-Net. DYPNet extracts features at five scales, HRDFPN fuses them along a top-down path, and SIFH supplies task-specific features to the classification and regression towers.

3.1 Backbone network

Shallow layers preserve small sign details that can be lost during downsampling. Fixed dense convolution, as used by conventional backbones [24, 25], samples the same relative positions regardless of the input. Modulated deformable convolution [26] computes the output at p_0 as

$$y_{(p_0)} = \sum_{k=1}^{K} w_k \cdot m_k \cdot x(p_0 + p_k + \Delta p_k) \quad (1)$$

where p_k is the kth kernel sampling position, w_k its weight, Δp_k the learned offset, and $m_k \in [0, 1]$ the learned modulation scalar.

D2KSC combines input-dependent kernel generation with adaptive sampling offsets. As shown in Fig. 3, a stride-one 1×1 convolution expands the channels, after which the features are split into a skip branch and a DYbottleneck branch. The branches are concatenated and projected by another stride-one 1×1 convolution. The skip path retains low-level information, while the sampled branch adapts to local shape and appearance. Figure 4 illustrates the difference between a regular grid and deformable sampling near a traffic sign.

3.2 Neck network

Bidirectional fusion, such as PANet [27], repeatedly aligns feature resolutions through upsampling and downsampling. In the examples in Fig. 5, shallow YOLO11 features preserve more sign-related

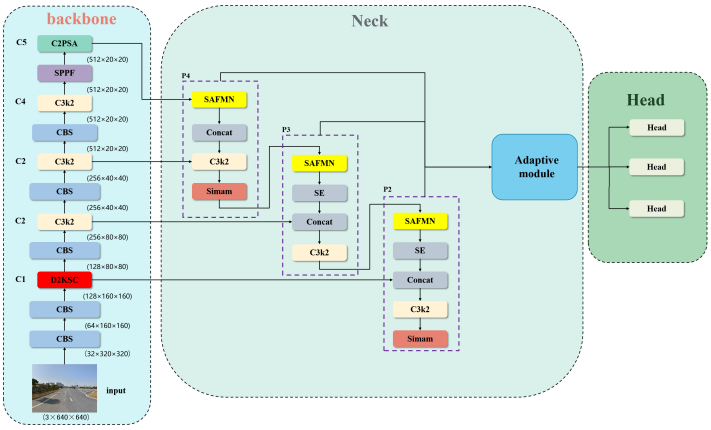

Fig. 2 Architecture of TSD-Net

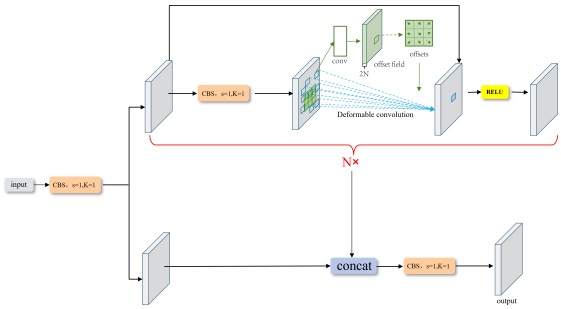

Fig. 3 Structure of the D2KSC feature extraction module. N denotes the number of repeated DYbottleneck blocks, and $N = 3$ is used in this study

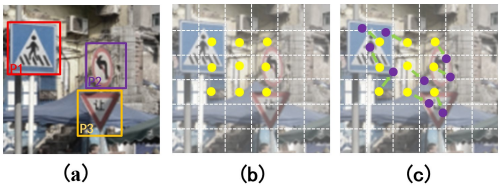

Fig. 4 Standard and deformable 3×3 sampling: (a) input, (b) regular grid, and (c) learned sampling offsets

detail than deeper features at the same 40×40 and 20×20 resolutions. These observations motivate a top-down neck that omits the bottom-up downsampling path.

The SAFMN upsampling block (Fig. 6) builds on spatially adaptive feature modulation [28]. Its attention block splits the normalized input X along the channel dimension and processes the four groups as follows:

$$[X_0, X_1, X_2, X_3] = \text{Split}(X) \tag{2}$$

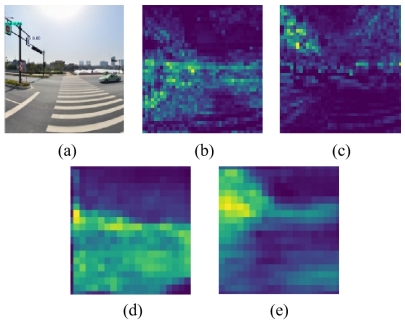

Fig. 5 Layer features: (a) input, (b,c) deep and shallow at 40×40, and (d,e) deep and shallow at 20×20

$$\hat{X}_0 = \text{DW-Conv}_{3\times3}(X_0) \tag{3}$$

$$\hat{X}_i = \uparrow_p \left(\text{DW-Conv}_{3\times3}\left(\downarrow_{\frac{p}{2^i}} (X_i)\right)\right) \quad 1 \le i \le 3 \tag{4}$$

Here, $\text{DW-Conv}_{3\times3}$ denotes the 3×3 depthwise convolution. The operator $\downarrow_{p/2^i}$ reduces the spatial size to $p/2^i$, and $\uparrow_p$ restores size p by nearest-neighbor interpolation.

At an input size of 640×640, HRDFPN provides 160×160, 80×80, and 40×40 maps at P2, P3, and P4, respectively. We set the output channel widths using

$$C_i = \lambda_i C_{\text{ref}}, \qquad i \in \{1, 2, 3\}, \tag{5}$$

with $C_{\text{ref}} = 512$. The selected factors $(0.50, 0.25, 0.50)$ give 256, 128, and 256 channels at P2, P3, and P4. Squeeze-and-Excitation recalibrates each of the three upsampled maps before fusion [29]. SimAM [30] is applied to the P4 feature continuing toward P3, and to the P2 feature before its concatenation with the shallow backbone

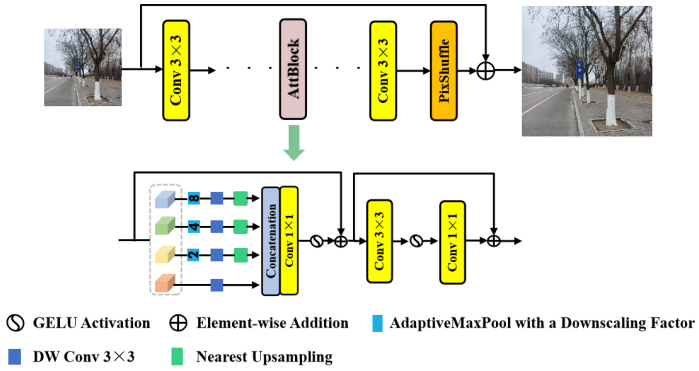

Fig. 6 Structure of the SAFMN

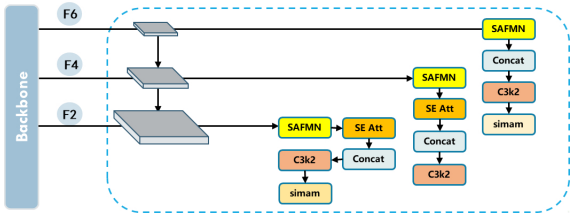

Fig. 7 Top-down feature fusion in HRDFPN

feature. The detection head receives the P4 feature before SimAM. Figure 7 shows the main fusion paths.

3.3 Detection head

SIFH (Fig. 8) uses features F_2, F_3, and F_4 at strides 4, 8, and 16. Classification adds an aligned and upsampled deeper feature, whereas localization concatenates the current feature with downsampled F_2:

$$F_i^{\text{cls}} = \begin{cases} F_i + U_2(A_{i+1\to i}(F_{i+1})), & i \in \{2,3\}, \\ F_4, & i = 4, \end{cases}$$
$$F_i^{\text{loc}} = \begin{cases} F_2, & i = 2, \\ \text{Cat}(F_i, D_{2^{i-2}}(F_2)), & i \in \{3,4\}. \end{cases} \quad (6)$$

Here, $A_{i+1\to i}$ is a 1×1 channel projection, U_2 is nearest-neighbor upsampling by two, and Cat concatenates channels. D_2 and D_4 use one and two stride-two depthwise separable convolution blocks, respectively. Separate towers process the resulting features.

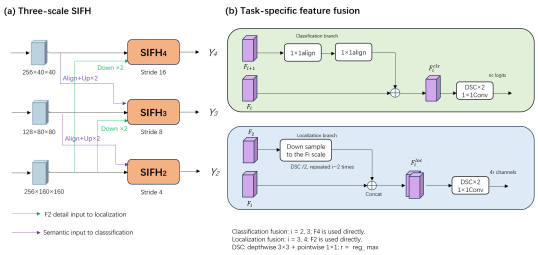

Fig. 8 The structure of SIFH

3.4 Loss

The objective combines classification and box regression:

$$L_{\text{total}} = L_{\text{cls}}(F^{\text{cls}}, \text{class}) + \lambda L_{\text{loc}}(F^{\text{loc}}, \text{box}) \quad (7)$$

The adaptive threshold focal loss [31] weights hard and easy examples using a threshold of 0.5:

$$\text{ATFL} = \begin{cases} -(\lambda - p_t)^{-\ln(p_t)} \log(p_t) & p_t \le 0.5 \\ -(1 - p_t)^{-\ln(\hat{p}_c)} \log(p_t) & p_t > 0.5 \end{cases} \quad (8)$$
$$\gamma = -\ln(\hat{p}_c) \quad (9)$$

For hard examples, the exponent depends on p_t; for easy examples, it depends on the class confidence $\hat{p}_c$. The latter combines the current target probability with historical epoch-average probabilities $\overline{p_i}$:

$$\hat{p}_c = 0.05 \times \frac{1}{t-1} \sum_{i=0}^{t-1} \overline{p_i} + 0.95 \times p_t \quad (10)$$

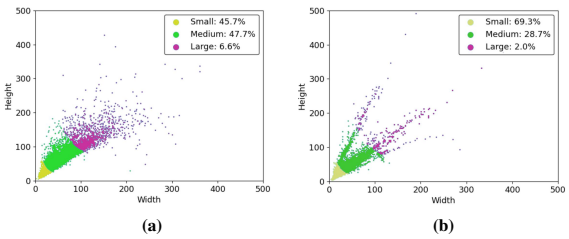

Fig. 9 Object-size distributions in (a) TT100K and (b) CCTSDB2021

For predicted positive-class probability p and label $y \in \{0, 1\}$, the target probability is

$$p_t = \begin{cases} p, & y = 1, \\ 1 - p, & y = 0. \end{cases} \tag{11}$$

The parameter λ is tunable, and γ changes with the class confidence.

4 Experiments

4.1 Implementation details

Datasets. For TT100K [32], we retain 45 categories with at least 50 instances following [33]. The processed split contains 5,962 training and 2,979 test images. CCTSDB2021 [34] provides 16,356 training images and 1,500 positive test images. Inputs are standardized to 640×640 RGB images. Figure 9 shows the object-size distributions using the COCO area thresholds; small and medium objects dominate both datasets.

Training and evaluation. The reported setup uses PyTorch 2.0.0, one NVIDIA RTX 4090 GPU with 24 GB of memory, and 300 epochs. Stochastic gradient descent uses momentum 0.937, weight decay 0.0005, and cosine annealing from 0.01 to 0.0001. We report precision, recall, F1, mean average precision at intersection over union 0.5 (mAP@50), and frames per second (FPS):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

$$\text{mAP50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_{50}^{i} \tag{15}$$

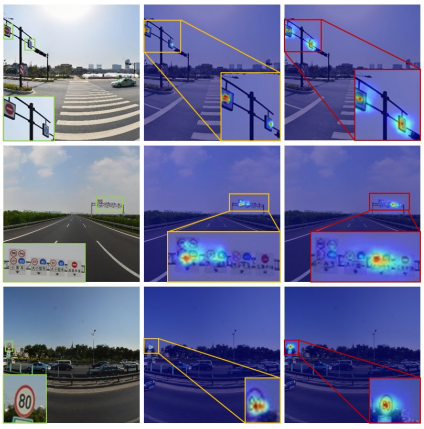

Fig. 10 Grad-CAM comparison: (a) inputs, (b) without HRDFPN, and (c) with HRDFPN

Here, TP, FP, and FN denote true positives, false positives, and false negatives, and N is the number of classes. Grad-CAM [35] visualizations with and without HRDFPN are shown in Fig. 10.

4.2 Performance comparison

On TT100K, TSD-Net achieves 93.7% mAP@50 with 7.8 million parameters and a reported throughput of 165 FPS (Table 1). Compared with YOLO-TS, this is an increase of 1.7 percentage points in mAP@50 and 20.4% in throughput. Figure 11 illustrates differences in small sign detections between YOLO11 and TSD-Net.

On CCTSDB, TSD-Net reports 96.4% mAP@50 and 83.3% recall with 8.2 million parameters (Table 2). Its mAP@50 is 0.2 percentage points above CAAD-Net and 0.1 below Libra R-CNN. Throughput values measured on different devices should be interpreted in the context of the listed hardware.

4.3 Ablation studies

Table 3 reports the module ablations with YOLO11 as the baseline. DYPNet alone improves mAP@50 from 82.2% to 83.1% and reduces the parameters from 25.8 to 23.2 million. HRDFPN alone improves mAP@50 by 9.4 percentage points; the Grad-CAM examples in Fig. 10 illustrate the resulting foreground responses. SIFH alone reduces the parameter count by 3.7 million. Combining SIFH with HRDFPN improves mAP@50 by 10.5 percentage points over the baseline. Using all three

Table 1 Performance comparison on TT100K dataset

Method	Precision(%)	Recall(%)	F1	Params(M)	mAP50(%)	FPS	GPU
TSR-SA [36]	-	-	-	-	90.2	48.8	V100
CDFF[37]	-	-	-	-	90.3	31.8	TITAN V
CAAD-Net[38]	71.2	78.8	-	51.8	90.3	10	RTX 2080Ti
RT-DETR-R18[39]	87	81	-	19.8	84.3	59.8	RTX 4090
YOLOv8-L[40]	83.4	77.2	0.802	43.6	84.7	71.4	RTX 4090
YOLOv9-C[41]	85.1	79	0.819	51.1	86.5	49.3	RTX 4090
YOLO11-L[42]	84.4	78.7	0.809	25.9	85.2	66.5	RTX 4090
YOLO-TS[12]	89.1	86.7	0.876	11.1	92.0	137.0	RTX 4090
Ours(TSD-Net)	92.1	86.9	0.88	7.8	93.7	165	RTX 4090

Table 2 Performance comparison on CCTSDB dataset

Method	Precision(%)	Recall(%)	F1	Params(M)	mAP50(%)	FPS	GPU
Libra R-CNN [43]	74.3	78.6	-	41.3	96.5	9.5	RTX 2080Ti
CAAD-Net [38]	78.8	80.7	-	51.8	96.2	10	RTX 2080Ti
UCN-YOLOv5 [44]	78.4	85.1	-	46.6	87.5	24.5	RTX 2080Ti
SignHRNet [45]	78.5	79.7	-	-	94.5	28	RTX 2080Ti
RT-DETR-R18 [39]	88.6	80.7	-	19.8	86	60.9	RTX 4090
YOLOv8-L [40]	89.0	76.6	0.823	43.6	84.3	90.9	RTX 4090
YOLOv9-C [41]	88.0	78.9	0.832	51.1	84.6	48.3	RTX 4090
YOLOv10-L [46]	88.5	75.9	0.817	25.8	82.2	76.4	RTX 4090
Ours(TSD-Net)	90.1	83.3	0.866	8.2	96.4	175.2	RTX 4090



(a) (b) (c) (d)

Fig. 11 TT100K examples: (a) inputs, (b) signs missed by YOLO11, (c) YOLO11 predictions, and (d) TSD-Net predictions

components gives 93.7% mAP@50 with 7.8 million parameters.

High-resolution control. Both P2-Control and HRD-FPN use P2/P3/P4 outputs at strides 4, 8, and 16. At epoch 300, HRD-FPN improves validation mAP@50 by 3.063 percentage points and mAP@50–95 by 2.900 points (Table 4). This comparison applies to the evaluated configurations.

Channel scaling. Table 5 compares eight channel allocations. Uniformly widening the three levels does not consistently improve accuracy. The selected $(0.50, 0.25, 0.50)$ setting gives the highest mAP@50, 91.6%, with 7.23 million parameters.

4.4 Inference speed

We measured inference on 1,500 TT100K validation images at batch size one. TSD-Net reaches 165 FPS on the RTX 4090. On Jetson Orin NX, TensorRT 8 with FP16 precision yields 46.2 FPS and 92.7% mAP@50 (Table 6).

5 Conclusion

TSD-Net combines adaptive sampling, high-resolution fusion, and task-specific prediction for small traffic signs. HRDFPN gives the largest

Table 3 Ablation results on the TT100K test set using YOLO11 as the baseline

Exp.	DYPNet	HRDFPN	SIFH	Precision(%)	Recall(%)	F1	Params(M)	mAP50(%)	FPS
YOLO11	–	–	–	88.9	76.9	0.837	25.8	82.2	76
1	✓			90.2	78.2	0.846	23.2	83.1	82
2		✓		91.6	83.3	0.863	14.1	91.6	124
3			✓	90.1	77.6	0.844	22.1	83.4	93
4	✓	✓		91.8	85.2	0.871	11.5	92.5	131
5		✓	✓	91.7	84.3	0.868	10.4	92.7	146
6	✓		✓	90.3	81.6	0.854	19.5	84.3	100
7	✓	✓	✓	92.1	86.9	0.88	7.8	93.7	165

Table 4 Validation performance of P2-Control and HRD-FPN at epoch 300

Method	P (%)	R (%)	F1	mAP_{50} (%)	$mAP_{50:95}$ (%)
P2-Control	86.639	79.685	0.8302	89.336	70.401
HRD-FPN	**90.361**	**83.437**	**0.8676**	**92.399**	**73.301**

Table 5 Channel scaling factors performance comparison

λ_1	λ_2	λ_3	Params(M)	mAP50(%)
0.25	0.25	0.25	6.27	87.9
0.50	0.50	0.50	7.62	91.1
0.25	0.50	0.50	6.94	90.1
0.25	0.25	0.50	6.60	89.7
0.50	0.25	0.25	6.90	88.6
0.50	0.25	0.50	7.23	**91.6**↑
0.50	0.50	0.25	7.28	89.9
0.25	0.50	0.25	6.62	88.2

Table 6 Performance comparison on TT100K on the mobile edge device

Method	mAP50(%)	FPS
YOLOv8-L	85.2	26.4
YOLOv9-C	85.6	23.2
YOLO11-L	84.3	25.5
ours	92.7	46.2

single-module mAP@50 gain, and the complete model improves the accuracy–size trade-off. Jetson Orin NX tests support edge deployment under the evaluated settings. Future work will assess driver assistance applications.

Statements and Declarations

Funding. This work was supported in part by the National Natural Science Foundation of China (Grants 62471428 and 82430040) and the Beijing Natural Science Foundation (Grant L2510005).

Competing interests. The authors declare that they have no relevant financial or non-financial interests to disclose.

Data availability. The original datasets are available from the Tsinghua–Tencent 100K project page [32] and the CCTSDB2021 repository [34].

Author contributions. Tianqi Liu conceived the methodology, designed and implemented the experiments, and drafted and polished the manuscript. Liyang Han analyzed data, prepared visualizations, and revised the manuscript. Shuo Hu directed and supervised the research, guided writing, and served as corresponding author. All authors reviewed and approved the final version.

References

[1] Liu, Y., Peng, J., Xue, J.H., Chen, Y., Fu, Z.H.: Tsingnet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild. Neurocomputing **447**, 10–22 (2021) https://doi.org/10.1016/j.neucom.2021.03.049

[2] Zaklouta, F., Stanciulescu, B.: Real-time traffic sign recognition in three stages. Robotics and Autonomous Systems **62**, 16–24 (2014) https://doi.org/10.1016/j.robot.2012.07.019 . New Boundaries of Robotics

[3] Yang, Y., Luo, H., Xu, H., Wu, F.: Towards real-time traffic sign detection and classification. IEEE Transactions on Intelligent Transportation Systems **17**, 2022–2031 (2016) https://doi.org/10.1109/TITS.2015.2482461

[4] Ayachi, R., Afif, M., Said, Y., Ben Abdelali, A.: An edge implementation of a traffic

sign detection system for advanced driver assistance systems. International Journal of Intelligent Robotics and Applications **6**, 207–215 (2022) https://doi.org/10.1007/s41315-022-00232-4

[5] Li, H., Sun, F., Liu, L., Wang, L.: A novel traffic sign detection method via color segmentation and robust shape matching. Neurocomputing **169**, 77–88 (2015) https://doi.org/10.1016/j.neucom.2014.12.111

[6] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015) https://doi.org/10.1038/nature14539

[7] Chen, Z., Yang, K., Wu, Y., Yang, H., Tang, X.: HCLT-YOLO: A hybrid CNN and lightweight transformer architecture for object detection in complex traffic scenes. IEEE Transactions on Vehicular Technology **74**(3), 3681–3694 (2025) https://doi.org/10.1109/TVT.2024.3496513

[8] Zhang, L., Yang, K., Han, Y., Li, J., Wei, W., Tan, H., Yu, P., Zhang, K., Yang, X.: TSD-DETR: A lightweight real-time detection transformer of traffic sign detection for long-range perception of autonomous driving. Engineering Applications of Artificial Intelligence **139**, 109536 (2025) https://doi.org/10.1016/j.engappai.2024.109536

[9] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 1137–1149 (2017) https://doi.org/10.1109/TPAMI.2016.2577031

[10] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91

[11] Aldoski, Z.N., Koren, C.: Traffic sign detection and quality assessment using yolov8 in daytime and nighttime conditions. Sensors **25**(4), 1027 (2025) https://doi.org/10.3390/s25041027

[12] Chen, J., Huang, H., Zhang, R., Lyu, N., Guo, Y., Dai, H.N., Yan, H.: YOLO-TS: Real-time traffic sign detection with enhanced accuracy using optimized receptive fields and anchor-free fusion. IEEE Transactions on Intelligent Transportation Systems **26**(11), 19995–20011 (2025) https://doi.org/10.1109/TITS.2025.3597710

[13] Liang, R., Qu, W., Li, S.: An adaptive multi-scale lightweight network for long-distance small traffic sign detection. Scientific Reports **16**(1), 13459 (2026) https://doi.org/10.1038/s41598-026-43920-y

[14] Zhang, Y., Bi, D., Huang, Q., Han, Y., Zhao, M.: A lightweight granular perception feature pyramid network with context-awareness for small traffic sign detection. Expert Systems with Applications **317**, 131885 (2026) https://doi.org/10.1016/j.eswa.2026.131885

[15] Li, M., Zhang, D., He, Q., Zhou, C., Li, M., Zhou, X.: MSD-YOLO: A multi-scale and detail-enhancement network for traffic sign detection. Computers, Materials & Continua **87**(3), 93 (2026) https://doi.org/10.32604/cmc.2026.076433

[16] Tian, C., Liu, Z., Liu, H., Yan, Y., Zhang, J.: Improved faster r-cnn traffic sign detection algorithm based on transformer. Journal of Physics: Conference Series **2637**, 012041 (2023) https://doi.org/10.1088/1742-6596/2637/1/012041

[17] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1951–1959 (2017). https://doi.org/10.1109/CVPR.2017.211

[18] Li, J., Deng, Q., Gao, W., Yang, B., Jia, L., Zhou, J., Pu, H.: DSF-YOLO for robust multiscale traffic sign detection under adverse weather conditions. Scientific Reports **15**(1), 24550 (2025) https://doi.org/10.1038/s41598-025-02877-0

[19] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid

networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106

[20] Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10778–10787 (2020). https://doi.org/10.1109/CVPR42600.2020.01079

[21] Liu, S., Huang, D., Wang, Y.: Learning spatial fusion for single-shot object detection. doi:10.48550/arXiv.1911.09516 (2019). https://arxiv.org/abs/1911.09516

[22] Hu, S., Tao, X., Zhao, X.: MCANet: Feature pyramid network with multi-scale convolutional attention and aggregation mechanisms for semantic segmentation. Journal of Visual Communication and Image Representation **110**, 104466 (2025) https://doi.org/10.1016/j.jvcir.2025.104466

[23] Tao, X., Hu, S.: AEDFPN: Efficient semantic segmentation via aggregation-enhancement-distribution mechanism. Information Sciences **729**, 122836 (2026) https://doi.org/10.1016/j.ins.2025.122836

[24] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[25] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243

[26] Xiong, Y., Li, Z., Chen, Y., Wang, F., Zhu, X., Luo, J., Wang, W., Lu, T., Li, H., Qiao, Y., Lu, L., Zhou, J., Dai, J.: Efficient deformable convnets: Rethinking dynamic and sparse operator for vision applications. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5652–5661 (2024). https://doi.org/10.1109/CVPR52733.2024.00540

[27] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018). https://doi.org/10.1109/CVPR.2018.00913

[28] Sun, L., Dong, J., Tang, J., Pan, J.: Spatially-adaptive feature modulation for efficient image super-resolution. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13144–13153 (2023). https://doi.org/10.1109/ICCV51070.2023.01213

[29] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018). https://doi.org/10.1109/CVPR.2018.00745

[30] Yang, L., Zhang, R.Y., Li, L., Xie, X.: SimAM: A simple, parameter-free attention module for convolutional neural networks. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 11863–11874 (2021). https://proceedings.mlr.press/v139/yang21o.html

[31] Yang, B., Zhang, X., Zhang, J., Luo, J., Zhou, M., Pi, Y.: EFLNet: Enhancing feature learning for infrared small target detection. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–11 (2024) https://doi.org/10.1109/TGRS.2024.3365677

[32] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2110–2118 (2016). https://doi.org/10.1109/CVPR.2016.232

[33] Zhang, L.J., Fang, J.J., Liu, Y.X., He, F.L., Rao, Z.Q., Zhao, J.X.: Cr-yolov8: Multiscale object detection in traffic sign images. IEEE Access **12**, 219–228 (2024) https://doi.org/10.1109/ACCESS.2023.3347352

[34] Zhang, J., Zou, X., Kuang, L.D., Wang, J., Sherratt, R.S., Yu, X.: Cctsdb 2021: A more comprehensive traffic sign detection benchmark. Human-centric Computing and Information Sciences **12**(23) (2022) https://doi.org/10.22967/HCIS.2022.12.023

[35] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017). https://doi.org/10.1109/ICCV.2017.74

[36] Chen, J., Jia, K., Chen, W., Lv, Z., Zhang, R.: A real-time and high-precision method for small traffic-signs recognition. Neural Computing and Applications **34**, 2233–2245 (2022) https://doi.org/10.1007/s00521-021-06526-1

[37] Wang, L., Wang, L., Zhu, Y., Chu, A., Wang, G.: Cdff: a fast and highly accurate method for recognizing traffic signs. Neural Computing and Applications **35**(1), 643–662 (2023) https://doi.org/10.1007/s00521-022-07782-5

[38] Wang, G., Zhou, K., Wang, L., Wang, L.: Context-aware and attention-driven weighted fusion traffic sign detection network. IEEE Access **11**, 42104–42112 (2023) https://doi.org/10.1109/ACCESS.2023.3264214

[39] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: DETRs beat YOLOs on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16965–16974 (2024). https://doi.org/10.1109/CVPR52733.2024.01605

[40] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. Accessed: 2025-11-11 (2023). https://github.com/ultralytics/ultralytics

[41] Wang, C.Y., Yeh, I.H., Liao, H.Y.M.: YOLOv9: Learning what you want to learn using programmable gradient information. In: Computer Vision – ECCV 2024, vol. 15089, pp. 1–21 (2024). https://doi.org/10.1007/978-3-031-72751-1_1

[42] Khanam, R., Hussain, M.: Yolov11: An overview of the key architectural enhancements. doi:10.48550/arXiv.2410.17725 (2024). https://arxiv.org/abs/2410.17725

[43] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 821–830 (2019). https://doi.org/10.1109/CVPR.2019.00091

[44] Liu, P., Xie, Z., Li, T.: Ucn-yolov5: Traffic sign object detection algorithm based on deep learning. IEEE Access **11**, 110039–110050 (2023) https://doi.org/10.1109/ACCESS.2023.3322371

[45] Yu, Y., Jiang, T., Li, Y., Guan, H., Li, D., Chen, L., Yu, C., Gao, L., Gao, S., Li, J.: SignHRNet: Street-level traffic signs recognition with an attentive semi-anchoring guided high-resolution network. ISPRS Journal of Photogrammetry and Remote Sensing **192**, 142–160 (2022) https://doi.org/10.1016/j.isprsjprs.2022.08.011

[46] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: YOLOv10: Real-time end-to-end object detection. In: Advances in Neural Information Processing Systems, vol. 37, pp. 107984–108011 (2024). https://doi.org/10.52202/079017-3429