2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DTS-YOLO: Enhancing Object Detection via Dynamic Routing, Texture Encoding, and Semantic Fusion

Shipeng Zheng*[0009-0005-3589-1928], Sen Zhang[0009-0008-7522-6270]

and Yulin Chen[0009-0001-4656-4990]

School of Software, Henan University, Kaifeng 475001, China
shipeng_zheng@henu.edu.cn

Abstract. To address limited feature representation, insufficient cross-scale fusion, and localization inaccuracies in complex scenes, we propose DTS-YOLO, a lightweight single-stage detector. It improves detection through dynamic feature aggregation, fine-grained texture encoding, and precise bounding box regression. Specifically, the Dynamic Route Enhanced Aggregation Module (DREAM) integrates multi-branch depthwise convolutions and lightweight Transformers to enrich multi-scale representations. To mitigate semantic inconsistency in fusion, Dynamic Cross-scale Feature Fusion (DCFF) combines Scale-aware Channel Attention Fusion (SCAF) and Intra-layer Feature Fusion Attention (IFFA) for enhanced semantic alignment. Additionally, edge and texture perception is reinforced via Sobel and Laplacian Pyramid modules. For robust localization, a novel Closed Complete IoU (CCIoU) loss introduces morphological closure operations to refine bounding box alignment under occlusion. Experiments on VisDrone2019 and DOTA-v1.5 (HBB) demonstrate consistent performance gains over baseline YOLO11, especially for small and dense objects in complex environments.

Keywords: Object detection, multi-scale fusion, texture encoding, semantic attention, bounding box regression.

1 Introduction

Object detection is a core task in computer vision, widely applied in autonomous driving, aerial surveillance, and industrial inspection. The YOLO series[1] has achieved strong performance by balancing speed and accuracy through advanced neck design and efficient backbones. However, YOLO11 still struggles in complex environments due to limited feature representation, suboptimal multi-scale fusion, and inaccurate localization under occlusion or crowding.

To overcome these issues, we propose DTS-YOLO, a fully enhanced detector that improves feature richness, semantic fusion, and localization accuracy. Our key contributions are as follows:

1. DREAM (Dynamic Route Enhanced Aggregation Module): Replaces YOLO11's C3k2 with multi-branch, re-parameterized convolutions and Transformer units to enlarge receptive fields and enrich representations, especially for small and complex targets.
2. DCFF (Dynamic Cross-scale Feature Fusion): Strengthens semantic consistency in top-down fusion using IFFA for intra-layer attention and SCAF for scale-aware cross-level fusion.
3. Edge and Texture Enhancement: Incorporates Sobel edges and Laplacian pyramids to improve low-level structure perception and robustness in texture-rich scenarios.
4. CCIoU Loss: Enhances box regression for dense, occluded, and rotated objects by simulating morphological alignment.

Extensive experiments on VisDrone and DOTA-v1.5 confirm that DTS-YOLO surpasses YOLO11 in both accuracy and robustness while maintaining real-time inference.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents our model architecture; Section 4 shows experiments; and Section 5 concludes the paper.

2 Related Work

2.1 Single-Stage Object Detectors

Single-stage detectors, represented by the YOLO series, are widely used for real-time detection due to their high efficiency. From YOLOv1 to YOLOv4 [2], and further to anchor-free models like YOLOv5 and YOLOX [3], performance has steadily improved. YOLO11 [4] enhances speed and accuracy via lightweight modules and refined post-processing. However, its backbone struggles with small targets and detailed regions in dense scenes. To address this, DTS-YOLO integrates DREAM, a dynamic multi-branch convolutional module with attention, to enhance feature expressiveness and lay the foundation for effective fusion and localization.

2.2 Multi-Scale Feature Fusion

Multi-scale fusion is essential for detecting objects of varying sizes. FPN and PANet use bidirectional paths to combine cross-layer features, while BiFPN [5] introduces weighted fusion to improve efficiency. However, YOLO11's shallow fusion neglects inter-layer semantic gaps. Attention-based methods offer improvements but lack dynamic semantic integration. To this end, DTS-YOLO introduces SCAF, which combines residual-aware semantic weighting with DREAM features to robustly integrate cross-scale information and improve performance in complex scenes.

2.3 Edge and Texture Enhancement

Edge and texture cues are critical for accurate boundary localization and small object detection. Traditional methods like Gabor, HOG, or HRNet [6] preserve details but are

computationally expensive. YOLO11, though efficient, underperforms in modeling fine textures. To compensate, DTS-YOLO incorporates lightweight Sobel edge and Laplacian pyramid modules [7], fusing them with backbone features before feeding into SCAF, thereby enhancing structural perception without significant overhead.

2.4 IoU-Based Localization Loss

IoU-based losses guide bounding box regression. While IoU and GIoU [8] rely on overlap, CIoU and DIoU [9] introduce distance and aspect penalties. However, YOLO11's CIoU is unstable in dense, occluded, or rotated scenarios. Though EIoU [10] aims to improve convergence, gains are limited. DTS-YOLO adopts CCIoU, which adds a morphological closure to CIoU, improving alignment in challenging scenes and enhancing robustness.

3 Method

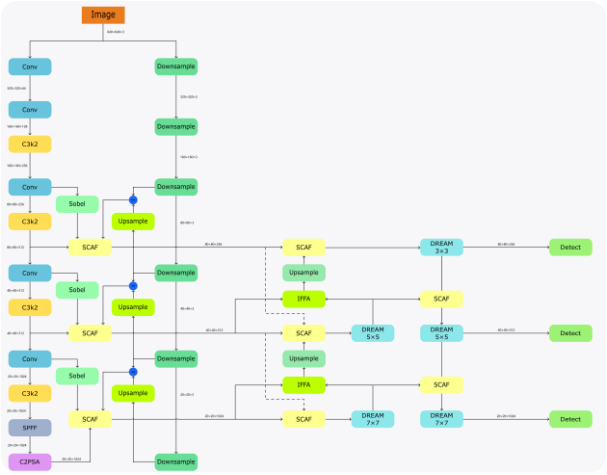
3.1 Overview of DTS-YOLO

The overall architecture of DTS-YOLO is shown in **Fig. 1**. The model enhances object detection through four key aspects: feature extraction, multi-scale fusion, detail perception, and precise localization. Compared to YOLO11, DTS-YOLO significantly improves detection accuracy in complex scenes while maintaining real-time performance. It also demonstrates strong applicability in aerial surveillance, dense crowd analysis, and industrial defect detection by addressing limitations in feature representation, semantic fusion, and boundary accuracy.

DREAM.
YOLO11's C3k2 module struggles with small targets and complex regions, especially under cluttered backgrounds. To address this, DREAM introduces a multi-branch dilated convolutional structure, combined with Layer Normalization (LN) [11] and Efficient Channel Attention (ECA) [12], to reduce inter-branch variation and improve downstream feature fusion. By enhancing key feature representations, DREAM significantly boosts detection accuracy for small objects, providing richer inputs for subsequent fusion modules.

DCFF.
YOLO11's shallow fusion strategy fails to capture cross-scale semantic relationships, limiting its multi-scale effectiveness. DCFF addresses this by introducing shallow features to guide deep semantics in the bottom-up path, enabling deeper layers to incorporate edge and texture cues from earlier stages. To achieve robust fusion, DCFF employs IFFA and SCAF to enhance intra- and inter-layer interactions, improving the adaptability and continuity of information flow across scales.

Fig. 1. Overall architecture of the proposed DTS-YOLO single-stage detector.



Edge and texture enhancement.
YOLO11 struggles to capture fine details, particularly edges and textures, in complex or highly detailed scenes. To address this, DTS-YOLO incorporates Sobel edge extraction [6] and Laplacian pyramid decomposition [7] into the early backbone stages, enhancing multi-scale detail representation and improving the network's sensitivity to low-level structural cues.

CCIoU.
In dense or occluded scenes, CIoU loss often fails to achieve precise alignment between predicted and ground-truth boxes. To address this, DTS-YOLO introduces CCIoU (Closed CIoU), which incorporates a morphological closure operation to fill small gaps and discontinuities. This enhances alignment accuracy, especially when box deviations are minor, and mitigates CIoU's limitations in guiding fine-grained localization.

3.2 DREAM: Dynamic Routing Enhanced Aggregation Module

To overcome YOLO11's limitations in multi-scale semantics and detailed feature modeling, we design DREAM, a lightweight, dynamic, and reparameterizable aggregation module that replaces the C3k2 block and forms the core of DTS-YOLO's semantic path. The structure of DREAM is shown in **Fig. 2**.

Limitations of C3k2 module in YOLO11.
YOLO11's C3k2 module employs two 3×3 convolutions with a residual connection [4], offering high inference efficiency due to its simplicity and low parameter count. However, its fixed receptive field, static fusion, and lack of multi-scale modeling hinder feature representation in the neck. While recent designs like RepHMS (e.g. MHAF-

YOLO [13]) adopt heterogeneous dilated branches to improve scale awareness, they introduce high complexity.

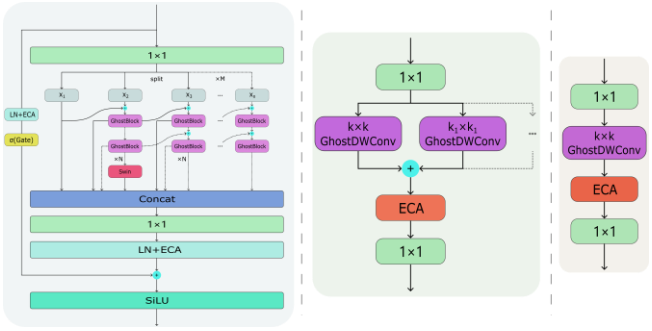


Fig. 2. Architecture of the DREAM module. Left: overall structure; Middle: GhostBlock (training); Right: GhostBlock (inference after reparameterization).

Multi branch structure and GhostDWConv.
To address fixed receptive fields and limited multi-scale modeling, the DREAM module uses a multi-branch architecture with different kernel sizes. To reduce computational cost, Depthwise Separable Convolutions[14](DWConv) are employed. Despite its lightweight nature, DWConv retains strong feature expressiveness.

To enhance efficiency, we use GhostDWConv, inspired by Ghost-Module, which generates ghost features using smaller kernels from the output of primary convolutions. This "main + ghost" strategy reduces redundancy and FLOPs while maintaining representational capacity. The main branch uses a 1×1 convolution to expand channels, while ghost features are generated via lightweight DWConvs to enrich feature diversity. Experiments show that a two-branch DREAM configuration reduces parameters by 36.49%, 30.33%, and 4.41% at 1024, 512, and 256 channels, respectively, while slightly improving detection performance.

Dynamic feature fusion
To address the static fusion limitations of C3k2, DREAM incorporates Layer Normalization (LN) [11], Efficient Channel Attention (ECA) [12], and a gated dynamic routing mechanism to enable adaptive feature fusion.

ECA enhances fusion accuracy by emphasizing informative channels. It captures global context via Global Average Pooling (GAP), then applies a lightweight 1D convolution followed by a Sigmoid function to generate per-channel attention weights. The final output is computed as:

$$Y_{fused} = Y_c \cdot \sigma\left(Conv1\times1\left(GAP(Y_c)\right)\right) \tag{1}$$

Among them, $\sigma(\cdot)$ represents the Sigmoid function, used to control the weight between 0 and 1.

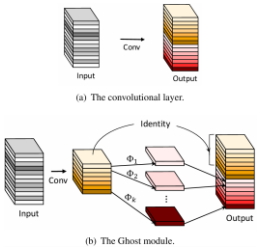


Fig. 3. Structural comparison between the Ghost module and standard convolutional layer (reproduced from [15])

Using ECA alone may cause training instability in multi-branch settings due to imbalanced activation across branches. To mitigate this, we apply Layer Normalization (LN) before ECA to stabilize feature distribution.

LN normalizes each channel to have zero mean and unit variance, ensuring consistent statistics across feature maps and promoting gradient stability. Specifically, LN computes the mean and standard deviation of each feature map and normalizes as follows:

$$Y_{\text{normalized}} = \frac{Y_c - \mu}{\sigma + \varepsilon} \tag{2}$$

Among them, μ and σ are respectively the mean and standard deviation, which ε are extremely small constants to prevent zero division errors.

The combination of LN and ECA allows adaptive channel weighting while preserving stable feature distributions, enhancing fusion robustness.

To further improve adaptability, we introduce a gate mechanism that dynamically fuses residual and backbone branches based on input features:

$$GATE\left(X_r\right) = \sigma\left(Conv1\times1\left(X_r\right)\right) \cdot X_r \tag{3}$$

The final output Y_{out} is the residual sum of weighted fused features and input features:

$$Y_{\text{out}} = Y_{\text{fused}} + \text{GATE}\left(X_r\right) \tag{4}$$

Among them, $\text{GATE}\left(X\right)$ denotes the gated feature map, $\sigma(\cdot)$ is the Sigmoid activation function, and X_r is the original input.

Lightweight Swin Tiny branch.

To enhance contextual modeling, DREAM integrates a lightweight Swin-Tiny branch at the end of its second path [16]. While conventional convolutions support multi-scale modeling, they are constrained by local receptive fields and struggle to capture global context—limiting their effectiveness in complex scenes.

Swin-Tiny enhances global context modeling via self-attention over shifted windows. In DREAM, the output of the second branch is passed through GhostDWConv, then fed into Swin-Tiny. The input is partitioned into fixed windows, where self-attention captures local dependencies. A window shift follows to enable cross-window interaction, allowing broader contextual aggregation. The resulting features are concatenated with the original input to enrich representation.

Given its computational cost, Swin-Tiny is applied only at the end of the second branch, where large-scale context is most needed and convolution alone is insufficient for long-range modeling. Experiments confirm that this design achieves optimal performance with minimal overhead.

Structural reparameterization.

During training, DREAM retains its multi-branch structure to enhance feature modeling. In inference, these branches are reparameterized into a single convolution via mathematically equivalent kernel fusion [16], significantly reducing computational complexity.

Let the three integral branches of a certain layer be fused by weighted summation:

$$Y_{\text{multi}} = \sum_{i=1}^{3} \alpha_i \cdot Y_i = \sum_{i=1}^{3} \alpha_i \cdot (W_i * X) \tag{5}$$

Among them, X is the input feature map, * representing the convolution operation.

In the inference phase, we eliminate the multi-path structure by expanding all kernels to the same size via zero-padding and summing them:

$$W_{\text{eq}} = \sum_{i=1}^{3} \alpha_i \cdot \text{Pad}(W_i) \tag{6}$$

Among them, $\text{Pad}(W_i)$ denotes zero-padding each kernel W_i a unified spatial size. The final fused kernel W_{eq} enables a single convolution to perform the equivalent operation of the original multi-branch design during inference.

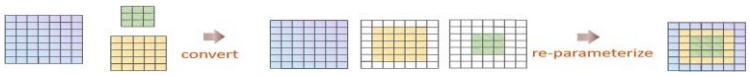


Fig. 4. Feature map conversion and reparameterization (reproduced from [16])

For structures with BatchNorm or LayerNorm, convolution and normalization layers can also be combined into an equivalent convolution kernel and bias term:

$$W_{\text{rep}} = \gamma \cdot \frac{W}{\sqrt{\sigma^2 + \varepsilon}}, \quad b_{\text{rep}} = \beta - \gamma \cdot \frac{\mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{7}$$

Among them γ, β, μ, σ^2 are the scaling parameters, bias, mean, and variance of the normalization layer, respectively, which W are the original convolution kernel weights.

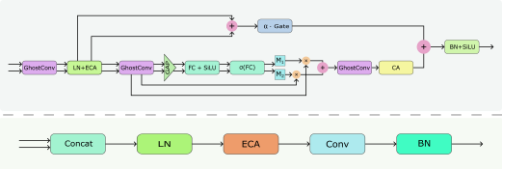


Fig. 5. Architectural schematic of the SCAF and IFFA modules (left: SCAF, right: IFFA)

3.3 Dynamic Cross Scale Feature Fusion (DCFF)

In YOLO11, feature fusion typically relies on naive concatenation or summation, which suffers from scale inconsistency and semantic misalignment. This can lead to key information loss and impaired performance on small objects.

To address these issues, we propose DCFF, which integrates SCAF, IFFA, and new top-down/bottom-up connections to enhance cross-scale semantic fusion and improve small object detection. To compensate for detail loss in high-level features, DCFF injects low-level cues—especially edges and contours—into deeper layers, improving both localization and classification accuracy.

We further fuse outputs from the backbone (containing rich high-level semantics) with refined representations from DREAM, which applies reparameterization to enhance feature selectivity. This complementary fusion strengthens the final feature map by combining raw contextual richness with dynamically optimized features.

SCAF（Semantic-aware Cross-scale Attention Fusion）

To address semantic misalignment in traditional multi-scale fusion, we propose SCAF, a refined module designed to align and enhance features across different resolutions.

Rather than using naive concatenation—which often leads to redundancy and poor integration of low-level spatial and high-level semantic features—SCAF employs a multi-stage attention-based fusion pipeline:

1. LN + ECA: Normalize channel distributions with Layer Normalization (LN), then apply Efficient Channel Attention (ECA) [12] to highlight informative features.
2. GAP + FC + SiLU: Extract global semantic cues via Global Average Pooling (GAP), followed by a fully connected layer and SiLU activation [18] to enhance feature expression.
3. Channel weighting via Gate: Use a learnable scalar α\alphaα and a Gate mechanism to adaptively control channel importance.
4. GhostConv + CA: Employ GhostConv for efficient feature extraction, and Channel Attention (CA) [30] to further refine channel-wise focus. ECA is placed at the input and CA at the output to balance local detail and global relevance.

By integrating hierarchical attention and dynamic weighting, SCAF improves feature expressiveness in cross-scale fusion, reduces redundancy, and outperforms traditional concatenation in efficiency and detection accuracy.

IFFA（Intra-layer Feature Fusion Attention）.

To improve feature integration before SCAF, we introduce IFFA, which adaptively fuses same-layer outputs from the backbone and DREAM modules.

These two sources contain distinct information: the backbone provides high-level semantics (e.g., object category), while DREAM refines low-level details (e.g., edges, small targets) via dynamic reparameterization. Direct fusion can result in redundancy or poor information alignment. IFFA applies weighted attention to balance their contributions and improve cross-source consistency.

Unlike typical pipelines using both BatchNorm (BN) and LayerNorm (LN) sequentially, which may cause over-adjustment and unstable flow, we apply LN+ECA for attention, followed by BN to reduce covariance shift and stabilize training.

Given that DREAM features are already highly expressive, applying nonlinear activations like SiLU may hinder rather than help. Thus, we omit SiLU in the final output fusion to preserve essential details.

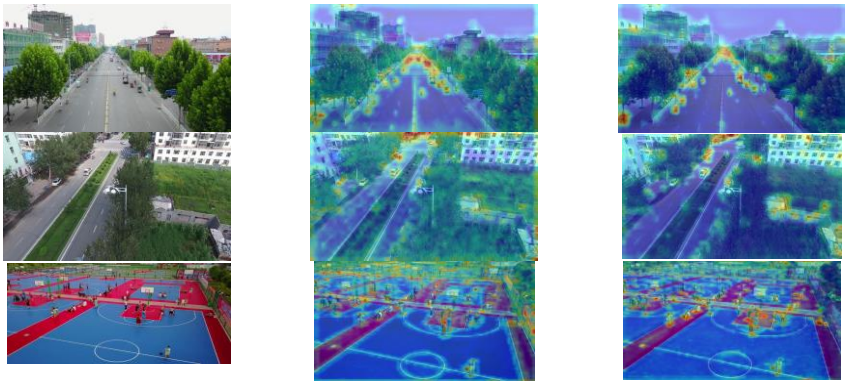
Experiments show that integrating IFFA before SCAF improves information quality and model performance with minimal parameter overhead.

DCFF (Dynamic Cross Scale Feature Fusion) significantly improves the performance of the DTS-YOLO model in complex environments through SCAF, IFFA, and innovative new connection methods, especially achieving excellent performance in small and multi-scale object detection. Through this new feature fusion strategy, DTS-YOLO achieves a good balance between detection accuracy and inference speed.

Table 1. Results of adding SCAF and IFFA on VisDrone2019 test set

Model	Precision	Recall	mAP@50	mAP@50-95
YOLO11n	45	33.2	32.3	19.2
YOLO11n+SCAF	46.1	33.7	33.1	19.5
YOLO11n+SCAF+IFFA	46.2	33.9	33.3	19.7

Fig. 6. Comparison of attention heatmaps. From left to right: original image, YOLO11 (baseline), and YOLO11+SCAF(with cross-scale fusion).



3.4 Edge and Texture Enhancement

Traditional YOLO architectures tend to prioritize global semantics, often neglecting fine-grained details of small objects—especially in complex scenes with large targets and cluttered backgrounds. To address this limitation in YOLO11, we embed the Sobel operator [6] and Laplacian pyramid [7] into the backbone to enhance local edge and texture features, improving robustness and detection accuracy.

Sobel operator

The Sobel operator enhances edge features such as contours and boundaries, thereby improving small object recognition. It uses two convolutional kernels to compute horizontal and vertical gradients.

While C3k2 outputs high-level semantics, it often lacks spatial detail due to repeated transformations. Applying Sobel directly on C3k2 is suboptimal. Instead, we apply Sobel to early convolutional outputs, where spatial details are preserved—especially beneficial for small object boundaries. Experiments confirm that fusing low-level edge maps with C3k2 features yields better detection performance.

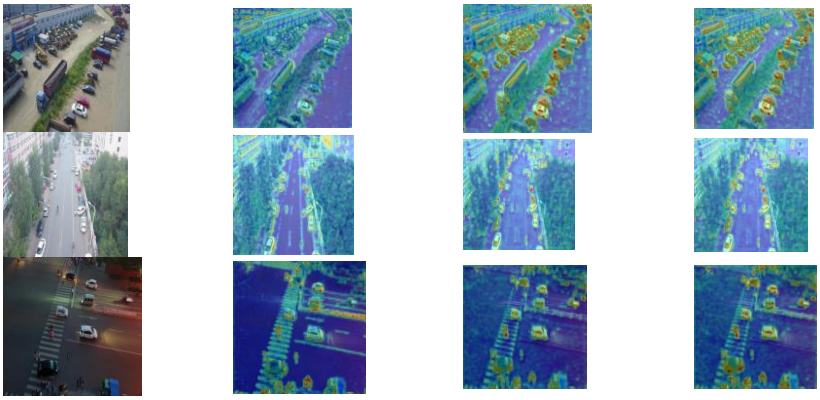


Fig. 7. Comparison of DTS-YOLO heatmaps with and without edge-texture enhancement. From left to right: original image, DTS-YOLO (without enhancement), DTS-YOLO (with enhancement), and difference map (highlighting the enhancement effect).

Laplacian pyramid

To improve multi-scale perception in DTS-YOLO, we integrate parallel Laplacian pyramids into the backbone. These pyramids decompose images into scale-specific representations, particularly enhancing edge and texture cues for small targets.

Instead of applying it to high-level backbone outputs, where detail may be lost, we directly process the original image to preserve spatial details across scales. This strategy improves edge precision and small object sensitivity with minimal overhead.

Combined with Sobel-based gradient enhancement, this non-parametric module enables fine-grained spatial modeling early in the backbone, complementing other

components in DTS-YOLO. Experiments confirm its effectiveness in enhancing robustness and accuracy in complex scenes.

Closed Complete IoU(CCIoU).

Localization loss plays a critical role in bounding box regression. While CIoU [9] incorporates center distance and aspect ratio penalties, it lacks explicit edge alignment constraints. In dense, occluded, or blurred scenes, CIoU provides weak guidance when predicted boxes are close to ground truth.

To address this, we propose CCIoU, which integrates a morphological closing operation into the CIoU framework to refine box alignment. Closing, composed of dilation followed by erosion, fills small gaps and smooths boundaries, producing more complete object coverage.

The specific calculation steps of CCIoU are as follows:

1. Inflation operation: Inflate the predicted box (b_1) to increase its size by a certain proportion. The expansion ratio is controlled by the hyperparameter CCIoU_ratio and the width and height of the prediction box:

$$\begin{aligned} dilated_{b_1x_1} &= x_1 - k_w,\ dilated_{b_1y_1} = y_1 - k_h \\ dilated_{b_1x_2} &= x_1 + k_w,\ dilated_{b_1y_2} = y_1 + k_h \end{aligned} \tag{8}$$

Among them, $k_w = \text{cliou\_ratio} \times w_1$, $k_h = \text{cliou\_ratio} \times h_1$

2. Calculate the intersection area between the expansion box and GT(b_2):

$$\begin{aligned} \text{inter\_x1} &= max\left(b_1x_1, b_2x_1\right),\ \text{inter\_x2} = min\left(b_1x_2, b_2x_2\right) \\ \text{inter\_y1} &= max\left(b_1y_1, b_2y_1\right),\ \text{inter\_x1} = min\left(b_1y_2, b_2y_2\right) \\ \text{inter\_area} &= (\text{inter\_x2} - \text{inter\_x1}) \times (\text{inter\_y2} - \text{inter\_y1}) \end{aligned} \tag{9}$$

Among them,

The coordinates b_1 of $\left(b_1x_1, b_1y_1\right)$ and $\left(b_1x_2, b_1y_2\right)$ represent the top left and bottom right corners of the box

The coordinates b_2 are $\left(b_2x_1, b_2y_1\right)$ and $\left(b_2x_2, b_2y_2\right)$

3. Calculate the union area after closure:

$$\text{union\_closing} = \left(dilated_{b_1x_2} - dilated_{b_1x_1}\right) \times \left(dilated_{b_1y_2} - dilated_{b_1y_1}\right) + w_2 \times h_2 - \text{inter\_area} \tag{10}$$

4. By calculating the intersection to union ratio between the expansion box and GT, obtain IoU_closing:

$$IoU_{closing} = \frac{\text{inter\_area}}{\text{union\_closing}} \tag{11}$$

5. Calculate closure loss:

$$\text{closing\_loss} = 1 - IoU_{closing} \tag{12}$$

6. The final CCIoU loss is the weighted sum of CIoU loss and closure loss:

$$CCIoU = 1 - CIoU + \alpha \times \text{closing\_loss} \tag{13}$$

Among them, α is a hyperparameter used to balance CIoU and closure loss

By introducing closure operations, CCIoU has improved boundary matching, enhanced detection accuracy for small targets, and improved localization accuracy, making the contact between boxes closer.

4 Experiments

4.1 Datasets

In this study, we used two publicly available object detection datasets: VisDrone2019 [20] and DOTAv1.5 [21].

1. VisDrone2019 dataset

A large-scale drone-captured dataset featuring complex aerial scenes and 10 object classes, including airplanes, ships, storage tanks, sports fields, and large vehicles. We follow the official split:

- Train: 5462 images, 42,168 annotations
- Val: 1143 images, 9357 annotations
- Test: Contains 1091 images, 8883 annotations

2. DOTAv1.5 dataset (HBB format)

To assess generalization on remote sensing tasks, we use DOTA-v1.5. Originally labeled with oriented bounding boxes (OBB), we convert the data to horizontal bounding boxes (HBB) for compatibility with mainstream detectors. It includes 16 object categories:

- Train: 5011 images, 22,592 annotations
- Val: 1449 images, 6244 annotations

4.2 Evaluation Metrics

We adopt standard object detection metrics to comprehensively evaluate model performance:

1. Precision

The proportion of predicted positives that are true positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

Among them, TP is True Positive and FP is False Positive.

2. Recall rate

The proportion of actual positives correctly predicted:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

Among them, FN is a false negative example.

3. mAP@50 (mean Average Precision at IoU=0.5)

Mean Average Precision at an IoU threshold of 0.5. It reflects detection accuracy per class, averaged across all categories.

4. mAP@50-95 (mean Average Precision at IoU=0.5 to IoU=0.95)

Mean Average Precision averaged over multiple IoU thresholds (from 0.5 to 0.95 with step 0.05). It provides a more robust assessment across varying levels of localization precision, especially for small objects and complex scenes.

These metrics jointly quantify model accuracy, localization robustness, and detection quality under diverse conditions.

4.3 Implementation Details

Our experiments are conducted on the Ultralytics framework using an NVIDIA RTX 3090 GPU and an Intel Xeon Gold 6330 CPU. Training runs for 300 epochs with cosine annealing and warmup scheduling. We adopt the Adam optimizer with momentum, use L2 regularization, and enable mixed precision training. Gradient accumulation simulates a batch size of 64.

We use Ultralytics' default data augmentation settings with 640×640 input resolution, disabling mosaic augmentation during the final 15 epochs for training stability.

4.4 Ablation Studies

We conduct ablation experiments on the VisDrone2019 test set to evaluate the contributions of individual modules. As edge and texture enhancement modules rely on SCAF for integration, and shallow-to-deep guidance is embedded in the neck, they are not ablated separately.

To ensure reliability, we report representative results consistent across multiple runs. **Table 2** shows that each module contributes positively, and the full DTS-YOLO configuration achieves the best performance in all metrics.

Table 2. Results of ablation experiments on VisDrone2019 test set

Model	Precision	Recall	mAP@50	mAP@50-95
YOLO11n	45	33.2	32.3	19.2
YOLO11n+DCFF	46.2	33.9	33.3	19.7
YOLO11n+DREAM	45.7	33.7	33.1	19.5
YOLO11n+CCIoU	45.1	33.8	32.7	19.4
DTS-YOLO	46.4	34.1	33.7	20.1

The results validate the effectiveness of each module and demonstrate that our integrated design significantly enhances robustness and detection performance.

4.5 Comparison with State-of-the-Art

We compare DTS-YOLOn with several state-of-the-art lightweight detectors on VisDrone2019 and DOTA-v1.5 (HBB format).

VisDrone2019 Results

As shown in **Table 3**, DTS-YOLOn achieves superior performance over YOLOv8n, RMVAD-YOLOn, YOLO11n, and other compact detectors. It reaches 33.7% mAP@50 and 20.1% mAP@50–95, demonstrating strong robustness in aerial scenes with small and multi-scale targets.

Table 3. Detection results on VisDrone2019 test set

Model	Precision	Recall	mAP@50	mAP@50-95
YOLOv8n[22]	37.9	28.8	26.3	14.5
YOLO-MMS[25]	51	40	27.16	-
RMVAD-YOLOn[22]	40.8	31	28.8	16.2
PG-YOLO[25]	38	43	31.6	-
BGF-YOLOv10[23]	-	-	32	-
YOLO11n	45	33.2	32.3	19.6
YOLO-ERF[24]	-	-	33.4	17.6
DTS-YOLOn	46.4	34.1	33.7	20.1

DOTA-v1.5-HBB comparative experiment

To assess generalization in remote sensing, we evaluate on DOTA-v1.5 (HBB). Although converting from OBB introduces localization simplification, it aligns with standard horizontal box constraints for real-world deployment.

Table 4. Detection results on DOTA-v1.5-HBB validation set

Model	Precision	Recall	mAP@50	mAP@50-95

ASF-YOLO[27]	-	-	61.3	38.9
TO-YOLOX[29]	-	-	63.02	-
YOLOv7-tiny[26]	-	-	64.9	35.5
YOLOv8n[26]	-	-	66.9	39.0
YOLO11n	76.9	63.2	68.9	45.8
DCEF^2-YOLO[28]	-	-	69.5	45.8
DTS-YOLOn	78.1	64.0	70.2	46.1

As shown in **Table 4**, DTS-YOLOn achieves 70.2% mAP@50 and 46.1% mAP@50–95, outperforming all baselines including YOLO11n, YOLOv8n, and DCEF2-YOLO. It also shows superior precision and recall, confirming its robustness in complex scenes.

5 Conclusion

We propose an efficient single-stage target detector, TDS-YOLO, which improves the detection accuracy and robustness while maintaining a lightweight structure, and solves the limitations in aspects such as L-feature representation, multi-scale fusion and positioning accuracy.

Extensive experiments on VisDrone2019 and DOTA-v1.5-HBB demonstrate DTS-YOLO's superior performance in detecting small objects and handling dense or complex textures, consistently outperforming the YOLO11 baseline.

Despite these achievements, challenges remain. Future work will focus on enhancing robustness under extreme conditions and improving the efficiency of multi-scale fusion. We also aim to extend DTS-YOLO to real-world applications such as autonomous driving and UAV-based monitoring to further assess its adaptability.

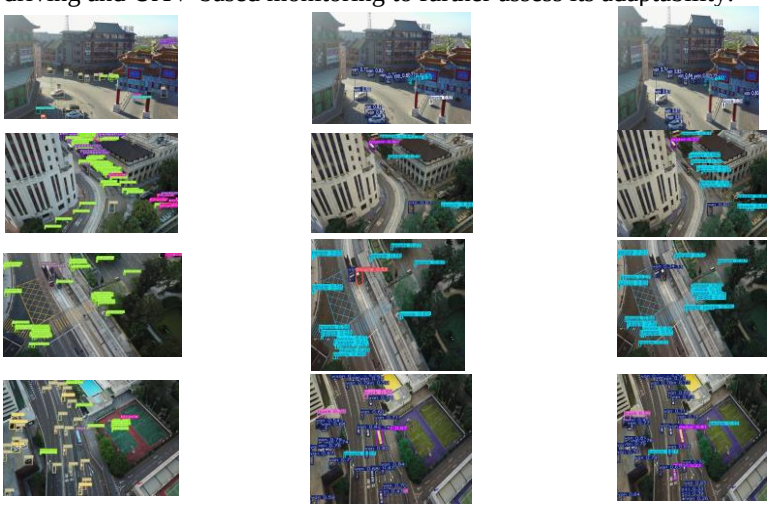


Fig. 8. Comparison of detection results among ground truth, YOLO11, and DTS-YOLO. From left to right: ground truth annotations, YOLO11 predictions, and DTS-YOLO predictions.

References

1. Hidayatullah, P., Syakrani, N., Sholahuddin, M.R., Gelar, T., Tubagus, R.: YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review, http://arxiv.org/abs/2501.13400, (2025). https://doi.org/10.48550/arXiv.2501.13400.
2. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection, http://arxiv.org/abs/2004.10934, (2020). https://doi.org/10.48550/arXiv.2004.10934.
3. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding YOLO Series in 2021, http://arxiv.org/abs/2107.08430, (2021). https://doi.org/10.48550/arXiv.2107.08430.
4. Khanam, R., Hussain, M.: YOLOv11: An Overview of the Key Architectural Enhancements, http://arxiv.org/abs/2410.17725, (2024). https://doi.org/10.48550/arXiv.2410.17725.
5. Tan, M., Pang, R., Le, Q.V.: EfficientDet: Scalable and Efficient Object Detection, http://arxiv.org/abs/1911.09070, (2020). https://doi.org/10.48550/arXiv.1911.09070.
6. 段瑞玲, 李庆祥, 李玉和: 图像边缘检测方法研究综述. 光学技术. 415–419 (2005).
7. Burt, P., Adelson, E.: The Laplacian Pyramid as a Compact Image Code. IEEE Trans. Commun. 31, 532–540 (1983). https://doi.org/10.1109/TCOM.1983.1095851.
8. Ye, N., Wolski, K., Mantiuk, R.K.: Predicting Visible Image Differences Under Varying Display Brightness and Viewing Distance. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5429–5437. IEEE, Long Beach, CA, USA (2019). https://doi.org/10.1109/CVPR.2019.00558.
9. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, http://arxiv.org/abs/1911.08287, (2019). https://doi.org/10.48550/arXiv.1911.08287.
10. Kim, J.: On the asymptotics of the shifted sums of Hecke eigenvalue squares, http://arxiv.org/abs/2011.06142, (2023). https://doi.org/10.48550/arXiv.2011.06142.
11. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer Normalization, http://arxiv.org/abs/1607.06450, (2016). https://doi.org/10.48550/arXiv.1607.06450.
12. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks, http://arxiv.org/abs/1910.03151, (2020). https://doi.org/10.48550/arXiv.1910.03151.
13. Yang, Z., Guan, Q., Yu, Z., Xu, X., Long, H., Lian, S., Hu, H., Tang, Y.: MHAF-YOLO: Multi-Branch Heterogeneous Auxiliary Fusion YOLO for accurate object detection, http://arxiv.org/abs/2502.04656, (2025). https://doi.org/10.48550/arXiv.2502.04656.
14. Zhang, P., Lo, E., Lu, B.: High Performance Depthwise and Pointwise Convolutions on Mobile Devices, http://arxiv.org/abs/2001.02504, (2020). https://doi.org/10.48550/arXiv.2001.02504.
15. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: GhostNet: More Features from Cheap Operations, http://arxiv.org/abs/1911.11907, (2020). https://doi.org/10.48550/arXiv.1911.11907.
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, http://arxiv.org/abs/2103.14030, (2021). https://doi.org/10.48550/arXiv.2103.14030.
17. Gu, R., Wang, G., Song, T., Huang, R., Aertsen, M., Deprest, J., Ourselin, S., Vercauteren, T., Zhang, S.: CA-Net: Comprehensive Attention Convolutional Neural Networks for Explainable Medical Image Segmentation. IEEE Trans. Med. Imaging. 40, 699–711 (2021). https://doi.org/10.1109/TMI.2020.3035253.

18. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning, http://arxiv.org/abs/1702.03118, (2017). https://doi.org/10.48550/arXiv.1702.03118.
19. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, http://arxiv.org/abs/1502.03167, (2015). https://doi.org/10.48550/arXiv.1502.03167.
20. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and Tracking Meet Drones Challenge, http://arxiv.org/abs/2001.06303, (2021). https://doi.org/10.48550/arXiv.2001.06303.
21. Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3974–3983. IEEE, Salt Lake City, UT (2018). https://doi.org/10.1109/CVPR.2018.00418.
22. Li, K., Zheng, X., Bi, J., Zhang, G., Cui, Y., Lei, T.: RMVAD-YOLO: A Robust Multi-View Aircraft Detection Model for Imbalanced and Similar Classes. Remote Sensing. 17, 1001 (2025). https://doi.org/10.3390/rs17061001.
23. Mei, J., Zhu, W.: BGF-YOLOv10: Small Object Detection Algorithm from Unmanned Aerial Vehicle Perspective Based on Improved YOLOv10. Sensors. 24, 6911 (2024). https://doi.org/10.3390/s24216911.
24. Wang, X., He, N., Hong, C., Sun, F., Han, W., Wang, Q.: YOLO-ERF: lightweight object detector for UAV aerial images. Multimedia Systems. 29, 3329–3339 (2023). https://doi.org/10.1007/s00530-023-01182-y.
25. Junos, M.H., Khairuddin, A.S.M.: YOLO-MMS for aerial object detection model based on hybrid feature extractor and improved multi-scale prediction. Vis Comput. (2024). https://doi.org/10.1007/s00371-024-03689-5.
26. Cui, C., Lv, F., Wang, R., Wang, Y., Zhou, F., Bian, X.: Research on Optical Remote Sensing Image Target Detection Technology Based on AMH-YOLOv8 Algorithm. IEEE Access. 12, 140809–140822 (2024). https://doi.org/10.1109/ACCESS.2024.3461337.
27. Lorian, V., Atkinson, B.: Effect of serum on gram-positive cocci grown in the presence of penicillin. J Infect Dis. 138, 865–871 (1978). https://doi.org/10.1093/infdis/138.6.865.
28. Shin, Y., Shin, H., Ok, J., Back, M., Youn, J., Kim, S.: DCEF2-YOLO: Aerial Detection YOLO with Deformable Convolution–Efficient Feature Fusion for Small Target Detection. Remote Sensing. 16, 1071 (2024). https://doi.org/10.3390/rs16061071.
29. Chen, Z., Liang, Y., Yu, Z., Xu, K., Ji, Q., Zhang, X., Zhang, Q., Cui, Z., He, Z., Chang, R., Sun, Z., Xiao, K., Guo, H.: TO–YOLOX: a pure CNN tiny object detection model for remote sensing images. International Journal of Digital Earth. 16, 3882–3904 (2023). https://doi.org/10.1080/17538947.2023.2261901.
30. Hou, Q., Zhou, D., Feng, J.: Coordinate Attention for Efficient Mobile Network Design, http://arxiv.org/abs/2103.02907, (2021). https://doi.org/10.48550/arXiv.2103.02907.