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.

Context-Aware Multi-Scale Detail-Preserving Fusion for Insulator Defect Detection

Linsihang Qiao

angelqiao93@gmail.com

State Grid Jibei Electric Power Company Limited Skills Training Center (Baoding Technical College of Electric Power)

Research Article

Keywords:

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11779607/v1

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

Additional Declarations: No competing interests reported.

Context-Aware Multi-Scale Detail-Preserving Fusion for Insulator Defect Detection

Linsihang Qiao

State Grid Jibei Electric Power Company Limited Skills Training Center
(Baoding Technical College of Electric Power), Baoding, Hebei, China.
Corresponding author(s). E-mail(s): angelqiao93@gmail.com;

Abstract

Insulator defect detection requires local detail to locate small damage and structural context to recognize missing components. We propose the Context-aware Multi-scale Detail-Preserving Network (CMDPNet), which connects these information sources before prediction. Partial-channel single-head attention models high-level context; a Gather-and-Distribute neck propagates multi-scale semantics; and Detail-Preserving Feature Fusion combines these semantics with shallow P2 features for four-scale prediction. On a fixed Transmission Line Insulation Defect (TLID) split, a 300-epoch cross-detector comparison achieves 62.6% mAP50-95, exceeding the best listed comparison by 1.1 percentage points, with AP75 higher by 3.0 percentage points. A separately reported 200-epoch, three-seed frozen test achieves 44.382%±0.602% mAP50-95, 5.250 percentage points above the same-group baseline. In our tests, a forward pass takes about 27 ms per image. These results support connecting context and high-resolution detail under the evaluated split, while quantifying the associated computational cost. The evidence remains limited to internal evaluation on one dataset; forward latency excludes the complete deployment pipeline.

Keywords: insulator defect detection; visual inspection; context modeling; multi-scale feature fusion; detail preservation

1 Introduction

Insulator defects in unmanned aerial vehicle (UAV) imagery require both local detail and structural context. Small damage may occupy only a few pixels, whereas missing structures must be recognized from the surrounding insulator shape. Scale variation and complex backgrounds complicate both tasks [1, 2].

Feature pyramid networks (FPNs) fuse semantics and location across levels [3], while DETR and RT-DETR use Transformer-based set prediction [4, 5]. However, high-level context does not recover shallow resolution, and high-resolution features also retain background texture. Our design connects context-enhanced semantics to shallow detail before prediction. Fig. 1 compares the resulting feature responses; independent normalization permits comparison of response locations and extent, but not absolute strength.

CMDPNet builds on ResNet-18 and RT-DETR, using SHViT's partial-channel single-head attention [6] and the Gather-and-Distribute mechanism [7]. Original P5 features enter global aggregation; context-enhanced Y5 features enter local fusion and deep aggregation. Detail-Preserving Feature Fusion (DPFF) then combines aggregated semantics with P2 detail for P2-to-P5 prediction. We evaluate this pathway through separately reported 300-epoch cross-detector comparisons and 200-epoch, three-seed frozen testing, together with unified forward-latency measurements.

Our contributions are:

1. We construct CMDPNet to connect high-level context modeling with multi-scale gathering and distribution for insulator defect detection.
2. We design DPFF to fuse aggregated $P3_{\text{out}}$ semantics with shallow P2 detail, supplying the high-resolution $P2_{\text{out}}$ branch for four-scale prediction.
3. We quantify accuracy and computational cost through structural ablation, three-seed validation and frozen testing, and unified complexity and forward-latency measurements.

2 Related Work

2.1 Insulator Defect Detection

Insulator defect detection uses locate-then-detect or direct pipelines [8]. Wen et al. suppress backgrounds using insulator masks, then apply Exact R-CNN, an improved Faster R-CNN [9, 10]. Direct methods improve features and computational cost through lightweight convolutions, reparameterized backbones,

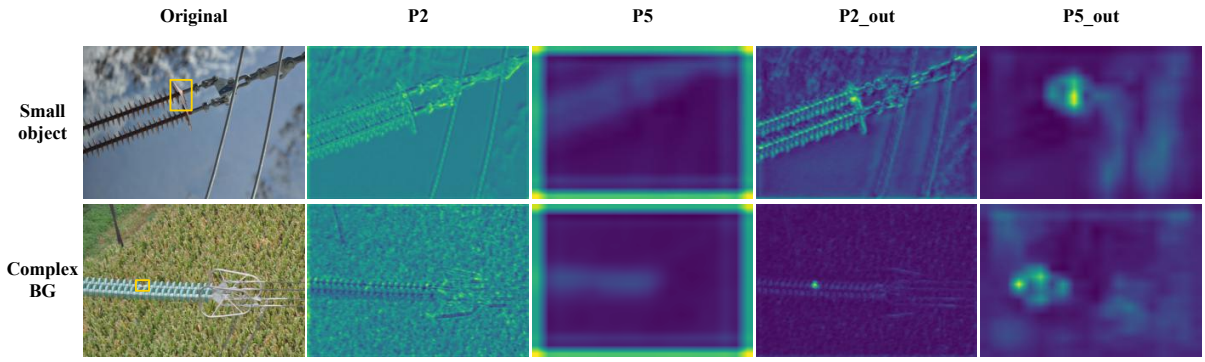


Fig. 1 Comparison of multi-layer feature responses under small-object and complex-background scenes. P2 and P5 denote the backbone's shallow and deep features, and $P2_{\text{out}}$ and $P5_{\text{out}}$ denote the corresponding outputs after multi-scale enhancement; each response map is independently normalized, comparing only the location and extent of high-response regions.

and denoising [11–13].

Against complex backgrounds, improved YOLOv8 combines attention and lightweight necks [1, 14], while RMC-YOLO combines long-range spatial information with multi-scale fusion [2]. YOLOv8-ASF-P2 uses ASF, a P2 detection layer, and pruning [15]; YOLOv8s-SwinT uses Swin, improved BiFPN, and a high-resolution small-object detection layer [16]. P2 detection is established; we study its link to high-level context and cross-scale semantics on the fixed TLID split [17].

2.2 Efficient Context and End-to-End Feature Modeling

Transformers use self-attention to model sequence relations [18]; ViT uses image-patch sequences [19]; Swin controls cost through hierarchy and shifted windows [20]. DETR uses end-to-end set prediction and bipartite matching [4]; Deformable DETR adds sparse multi-scale sampling [21]; DINO improves denoising training and query initialization [22]. RT-DETR targets real-time detection using efficient hybrid encoding and query selection [5]; RT-DETRv3 adds hierarchical dense positive supervision during training [23].

2.3 Multi-Scale Feature Fusion and Detail Preservation

FPN links deep and shallow features top-down [3]; PANet adds a bottom-up path [24]; EfficientDet's BiFPN learns bidirectional fusion weights [25]. Gold-YOLO's Gather-and-Distribute gathers multi-layer information, then distributes global representations across scales [7].

SE models channel dependencies [26]; CBAM applies channel then spatial attention [27]. DPFF fuses P2 with aggregated semantics to link context modeling and high-resolution prediction.

3 Method

3.1 Overall Framework

CMDPNet combines structural context and local detail (Fig. 2). ResNet-18 extracts P2–P5 [28], spanning high-resolution detail to high-level semantics.

3.2 Context-Aware SHSA Encoder

After a 1×1 convolution projects P5 to 256 channels, SHSA [6] applies spatial attention to one quarter and bypasses the remainder (Fig. 3).

Input $X \in \mathbb{R}^{C\times H\times W}$ splits into attention X_a and bypass X_b branches:

$$\begin{aligned} X &= \mathrm{Cat}(X_a, X_b), \quad C_a = \lfloor C/4 \rfloor, \\ C_b &= C - C_a, \quad d_q = d_k = \lfloor C/2 \rfloor, \quad d_v = C_a. \end{aligned} \tag{1}$$

Group-normalized X_a maps to queries Q, keys K, and values V, flattened as Q, K $\in \mathbb{R}^{d_k\times N}$ and V $\in \mathbb{R}^{d_v\times N}$.

$$[Q, K, V] = \mathrm{Split}\big(\phi_{qkv}(\mathrm{GN}(X_a))\big), \quad N = HW. \tag{2}$$

Softmax row-normalizes query–key similarities over key positions:

$$A = \mathrm{Softmax}\left(Q^T K/\sqrt{d_k}\right). \tag{3}$$

The weighted $VA^{\mathsf{T}} \in \mathbb{R}^{d_v\times N}$ is reshaped and concatenated with X_b; ψ projects the resulting $d_v + C_b$ channels back to C (Eq. (4)).

$$\begin{aligned} X'_a &= \mathrm{reshape}(VA^T), \\ \mathrm{SHSA}(X) &= \psi\big(\mathrm{Cat}(X'_a, X_b)\big). \end{aligned} \tag{4}$$

Residual updates use layer normalization and a convolutional feed-forward network with mappings W_1

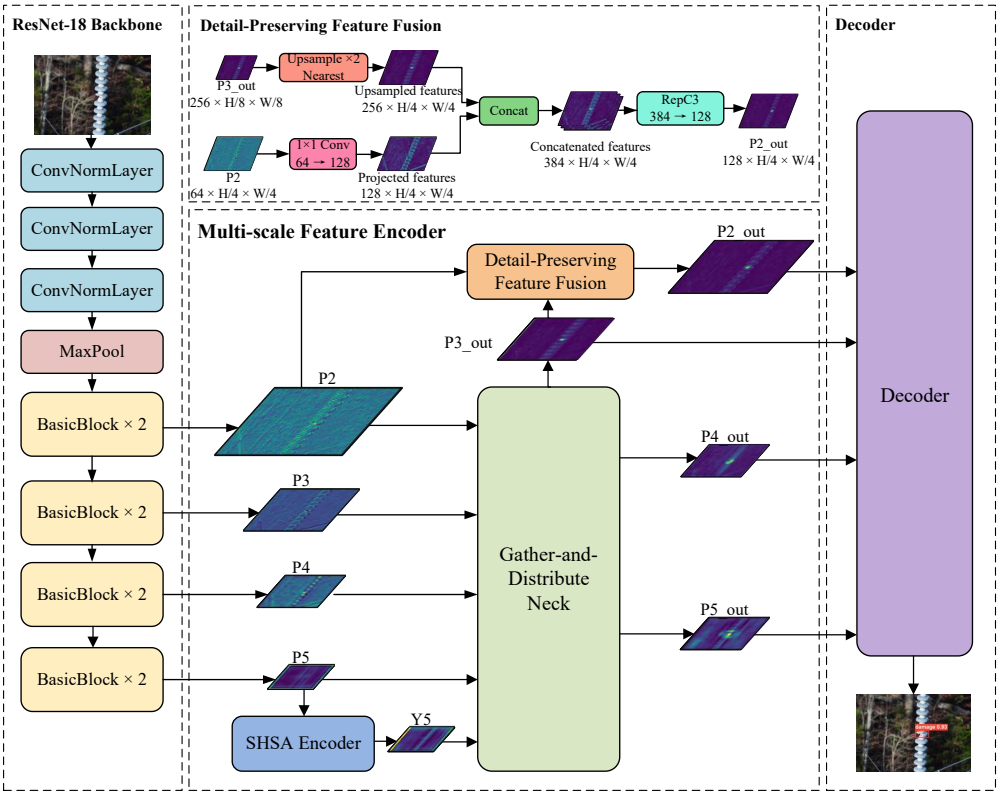


Fig. 2 CMDPNet overview. SHSA supplies P5 context to multi-scale fusion, while DPFF adds a P2-resolution branch for four-scale prediction.

and W_2 (Eq. (5)). A final 1×1 projection ϕ_o outputs Y5. Dropout is zero and omitted.

$$\begin{aligned} Z &= \mathrm{LN}(X + \mathrm{SHSA}(X)), \\ Y_5 &= \phi_o\big(\mathrm{LN}(Z + W_2\,\mathrm{GELU}(W_1 Z))\big). \end{aligned} \tag{5}$$

3.3 Gather-and-Distribute Multi-Scale Feature Aggregation

Gather-and-Distribute [7] injects aggregated multi-scale features into local branches (Fig. 4).

Low-GD concatenates original P2–P5 (including backbone F_{P5}) at P4 resolution; the Information Fusion Module (IFM) splits the result into P4/P3 global groups.

$$\begin{aligned} F_G^L = \mathrm{Cat}\big(&\mathrm{Pool}(F_{P2}), \mathrm{Pool}(F_{P3}), \\ &F_{P4}, \mathrm{Up}(F_{P5})\big). \end{aligned} \tag{6}$$

$$[G_4^L, G_3^L] = \mathrm{Split}\big(\phi_{\mathrm{IFM}}(F_G^L)\big). \tag{7}$$

InjectionMultiSum combines local feature F_l with global group G_j (Eq. (8)). R_j aligns spatial resolution; ϕ_l, ϕ_a, and ϕ_g map local content, gate, and global content. The hard-sigmoid gate σ_h modulates local content before global addition.

$$\begin{aligned} F_o = \phi_l(F_l) \odot \sigma_h\big(R_j(\phi_a(G_j))\big) \\ + R_j(\phi_g(G_j)). \end{aligned} \tag{8}$$

P4 local fusion also receives Y5. RepC3 reorganizes the injected features into intermediate P4 and $P3_{\mathrm{out}}$. High-GD aligns these with Y5 through PyramidPoolAgg, then applies TopBasicLayer (TBL) for global modeling.

Each F_i is adaptively pooled to $H' \times W'$, determined by reference input F_n and stride s (Eq. (9)). Here, H_n, W_n are the reference dimensions, $W' = \lfloor (W_n - 1)/s \rfloor + 1$, and ϕ_p projects concatenated features. The TBL output is channel-mapped and split for P4/P5 injection.

$$\begin{aligned} F_p &= \phi_p\left(\mathrm{Cat}_{i=1}^{n}\,\mathrm{Pool}_{H',W'}(F_i)\right), \\ H' &= \lfloor (H_n - 1)/s \rfloor + 1. \end{aligned} \tag{9}$$

$$[G_4^H, G_5^H] = \mathrm{Split}\big(\phi_{1\times1}(\mathrm{TBL}(F_p))\big). \tag{10}$$

Injecting these groups into P4/P5 produces $P4_{\mathrm{out}}/P5_{\mathrm{out}}$.

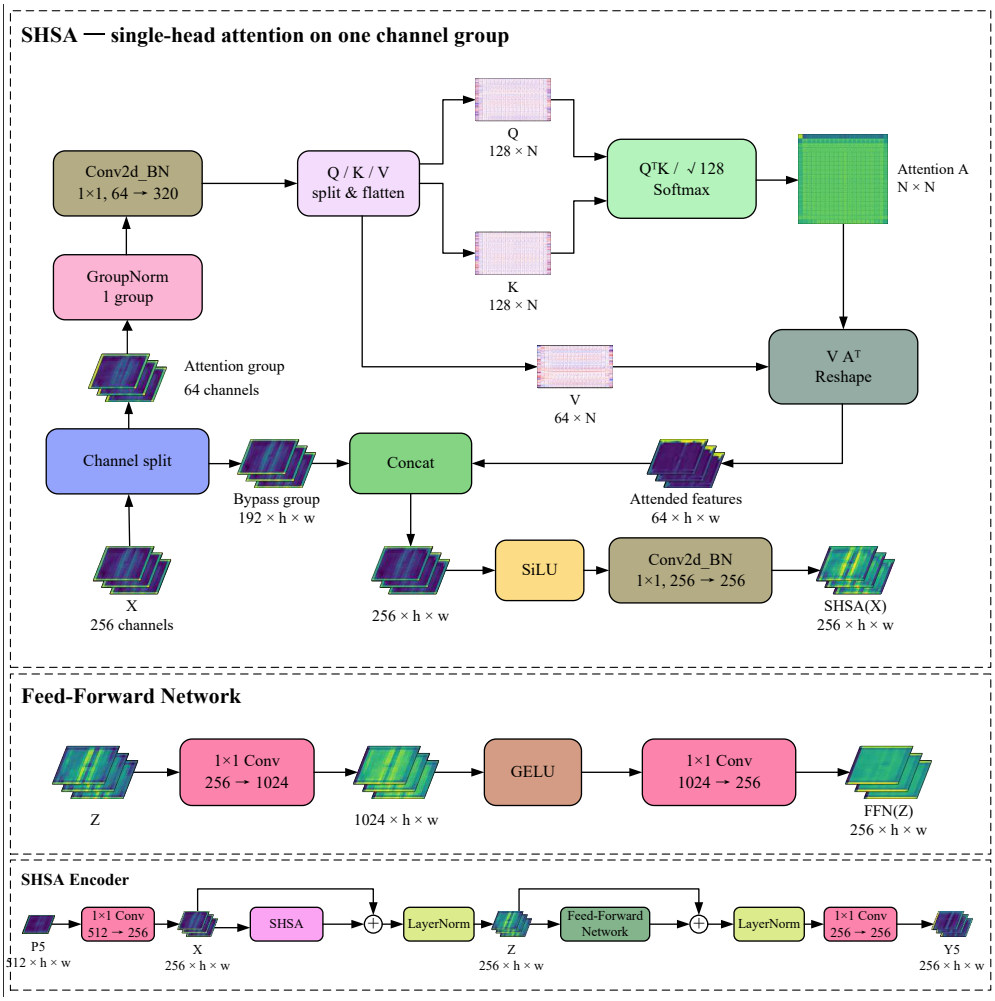


Fig. 3 SHSA Encoder: partial-channel attention (top), convolutional feed-forward network (middle), and encoder update (bottom). Attended and bypass channels are concatenated and projected before the feed-forward update.

3.4 Detail-Preserving Fusion and Four-Scale Prediction

The fused outputs reach only P3 resolution. DPFF adds a higher-resolution prediction branch by combining backbone P2 detail with $P3_{\text{out}}$ semantics (Fig. 5).

In Eq. (11), Up_2 applies 2× nearest-neighbor upsampling to $P3_{\text{out}}$, and $\phi_{1\times1}$ projects P2 channels. RepC3 processes their spatially aligned concatenation to produce $P2_{\text{out}}$.

$$F_{P2}^{\text{out}} = \text{RepC3}\big(\text{Cat}\big(\text{Up}_2(F_{P3}^{\text{out}}), \phi_{1\times1}(F_{P2})\big)\big). \tag{11}$$

The four-scale Transformer decoder performs classification and bounding-box regression:

$$\mathcal{F}_{\text{dec}} = \{F_{P2}^{\text{out}}, F_{P3}^{\text{out}}, F_{P4}^{\text{out}}, F_{P5}^{\text{out}}\}. \tag{12}$$

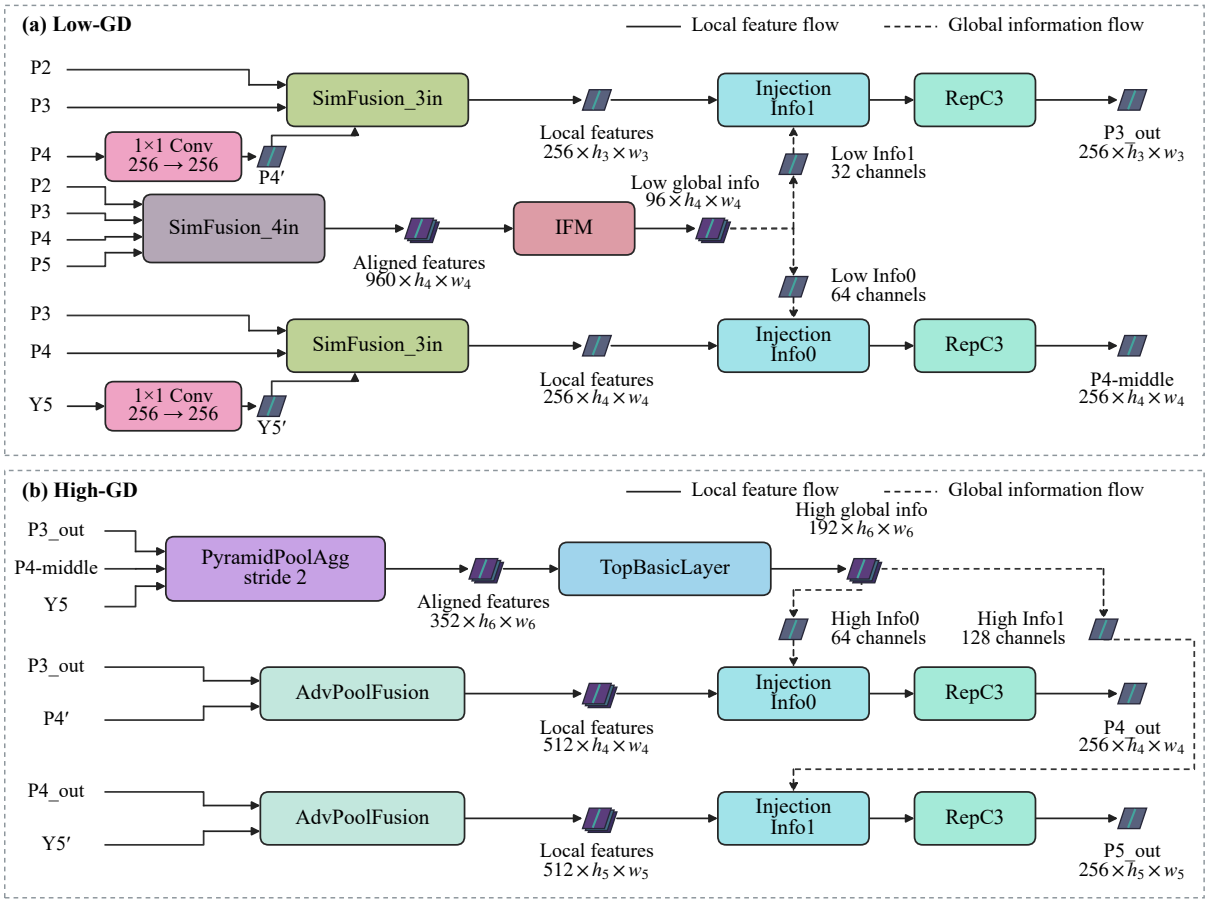


Fig. 4 Gather-and-Distribute aggregation: (a) Low-GD; (b) High-GD. Dashed and solid lines denote global information flow and local feature flow, respectively.

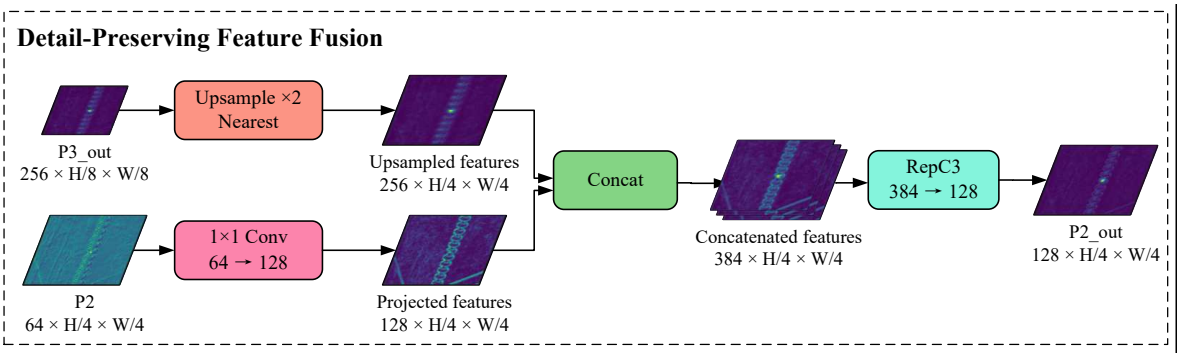


Fig. 5 DPFF and four-scale prediction. Upsampled $P3_{\text{out}}$ semantics and projected P2 detail form $P2_{\text{out}}$ through RepC3; all four scales enter the Transformer decoder.

4 Experiments and Analysis

4.1 Dataset and Experimental Setup

TLID [17] contains 712 images and 816 damage/drop boxes, with a shared file-level split of 499/106/107 training/validation/test images (Table 1). Group A compares detectors; Group B evaluates ablations and multiple seeds before frozen testing. Their different budgets and evaluation stages preclude matched-condition comparison.

Group A/B use 300/200 epochs and AdamW [29] (Table 2). For each architecture evaluated in both groups,

Table 1 TLID frozen data split

Split	Images	Annotated instances		
		damage	drop	Total
Train	499	258	318	576
Val	106	47	73	120
Frozen test	107	46	74	120
Total	**712**	**351**	**465**	**816**

Note: splits are mutually exclusive at the file level; the test set is an internal frozen hold-out.

preprocessing, architecture configuration, pre-trained weights, and evaluation code are identical. Within Group B, only structure and the pre-specified seed vary.

CMDPNet and RT-DETR use focal/L1/GIoU Hungarian matching [4] (Eq. (13)) and Varifocal [30]/L1/GIoU [31] losses (Eq. (14)). Other detectors retain native matching and losses under the shared budget and evaluation procedure (Sect. 4.2).

$$C_{\text{match}} = 2C_{\text{cls}}^{\text{FL}} + 5\|\hat{b} - b\|_1 + 2\big(1 - \text{GIoU}(\hat{b}, b)\big). \tag{13}$$

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{VFL}} + 5\mathcal{L}_1 + 2\mathcal{L}_{\text{GIoU}}, \qquad \mathcal{L}_{\text{total}} = \sum_{\ell \in \Omega} \mathcal{L}_{\text{det}}^{(\ell)}. \tag{14}$$

The supervised branches Ω comprise encoder outputs, auxiliary decoder layers, and denoising predictions, all using the same detection loss.

4.2 Evaluation Metrics

Both groups share AP statistics and matching, with architecture-specific post-processing. AP50/AP75 use IoU thresholds 0.50/0.75; mAP50-95 equally averages ten COCO-style thresholds [32], 0.50:0.05:0.95, and both classes. AP uses the interpolation/integration procedure below. Precision (P) and recall (R) are measured at the maximum smoothed, class-averaged F1. Complexity is measured by parameters, GFLOPs, and mean forward latency.

RT-DETR variants directly resize inputs to 640 × 640 and rank all 300 queries by each query's highest class score, without confidence clipping or NMS. NMS-based convolutional detectors use confidence 0.001, class-wise NMS IoU 0.70, and at most 300 boxes/image. The shared evaluator sorts same-class pairs satisfying each IoU threshold by IoU, then de-duplicates predictions and ground truths for one-to-one matching.

For predicted/ground-truth boxes B_p/B_g and their smallest enclosing box C, IoU, GIoU, P, and R are

Table 2 Group A/B protocols and the unified efficiency evaluation

Setting	Protocol value
Input & split	640×640; same 499/106/107 file-level split
Training epochs	Group A: 300 epochs; Group B: 200 epochs
Batch size	4
Optimizer	AdamW
Initial LR & weight decay	Both 1×10^{-4}
AMP	Off (amp=False)
Workers	4
Initialization	COCO pre-training for the matching architecture; the same architecture uses the same source in both groups
Random seed	Group A: 0; Group B: selection 0, re-verification 0/1/2
Structure selection & weight choice	Group B selects structures by peak validation mAP50-95; both groups save locked weights by validation fitness
Unified evaluation	Unified AP integration and same-class one-to-one matching; RT-DETR keeps 300 queries with no confidence clipping or NMS; other architectures are specified separately
Efficiency protocol	RTX 4060; FP32; batch=1; 640×640; fused-model forward pass only
Timing setup	200 warm-up runs; 3 groups of 1000 runs; CUDA synchronized before/after each run

$$\text{IoU} = \frac{|B_p \cap B_g|}{|B_p \cup B_g|}, \qquad \text{GIoU} = \text{IoU} - \frac{|C \setminus (B_p \cup B_g)|}{|C|}. \tag{15}$$

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}. \tag{16}$$

For class c and IoU threshold τ, confidence-ranked TP/FP counts define precision–recall nodes. Recall endpoints 0/1 with precision sentinels 1/0 are added before a reverse cumulative maximum forms the envelope. Linear interpolation at $r_j = j/100$ ($j = 0, \ldots, 100$) and trapezoidal integration give AP (Eq. (17)); $\hat{P}(j/100)$ denotes the interpolated envelope.

$$\hat{P}(r) = \max_{\tilde{r} \geq r} P(\tilde{r}), \qquad AP_{c,\tau} = \frac{1}{100}\left[\frac{1}{2}\hat{P}(0) + \sum_{j=1}^{99} \hat{P}(j/100) + \frac{1}{2}\hat{P}(1)\right]. \tag{17}$$

Table 3 Group A cross-detector comparison

Model		Detection performance (%) ↑					Efficiency ↓		
Family	Variant	mAP50-95	AP50	AP75	Precision	Recall	Params (M)	FLOPs (G)	Lat. (ms)
YOLOv11	M	57.5	86.5	63.1	93.8	79.2	20.0	67.6	18.20
	L	58.1	87.0	64.0	93.7	79.4	25.2	86.6	25.94
	X	58.4	87.0	65.0	93.8	79.8	56.8	194.4	27.88
YOLOv10	M	55.1	85.0	61.2	94.0	76.2	**16.4**	63.4	17.84
	B	55.5	85.2	61.3	92.5	77.0	20.4	97.9	18.44
	L	55.2	85.0	61.1	92.2	78.0	24.4	120.3	20.87
	X	55.7	85.4	61.8	93.4	77.2	29.5	160.4	21.95
YOLOv9	M	58.1	85.8	66.0	93.4	77.9	20.0	76.3	18.84
	C	58.5	87.1	66.0	94.3	79.1	25.3	102.1	20.82
	E	59.1	88.1	66.8	93.9	80.1	57.3	189.0	21.91
GOLD-YOLO	M	57.0	86.2	64.5	93.5	77.6	41.3	87.5	17.20
	L	57.8	86.4	65.0	93.3	78.5	75.1	151.7	20.45
YOLOv8	M	58.2	86.8	65.1	94.3	79.0	25.9	79.3	14.11
	L	58.5	87.2	65.0	94.6	79.1	43.7	165.7	16.98
	X	59.4	88.0	67.1	94.2	79.2	68.2	257.8	17.55
DAMO-YOLO	M	58.0	86.6	65.2	93.1	79.0	28.2	61.8	19.14
	L	58.8	88.1	65.7	93.0	80.0	42.1	97.3	24.36
YOLOv6	M	53.5	83.5	60.1	93.3	74.9	34.9	85.8	**13.32**
	L	54.1	83.2	59.6	92.8	75.3	59.6	150.7	16.54
YOLOv5	M	56.8	85.6	63.9	93.6	77.2	25.0	64.0	16.42
	L	57.3	86.0	65.1	94.0	78.4	53.2	135.6	18.97
	X	58.0	86.8	66.0	93.4	78.5	97.2	247.3	24.05
YOLOv7	—	49.1	77.3	57.0	86.9	70.4	36.9	104.7	16.85
	X	49.6	77.8	57.5	93.0	68.8	71.3	189.9	18.74
YOLO-MS	M	52.4	79.0	59.2	82.9	68.1	22.2	80.2	30.85
RT-DETRv3	R18	57.1	88.4	65.0	95.5	82.8	20.0	60.0	21.33
	R34	59.3	90.5	66.0	96.1	85.1	31.0	92.0	22.01
	R50	61.5	92.6	68.4	96.6	87.2	42.0	136.0	29.41
RT-DETRv1	R18	56.9	88.5	64.8	95.8	83.4	19.8	**57.2**	24.96
	R34	57.7	88.8	65.1	96.8	84.5	29.9	87.4	28.11
	R50	59.8	90.7	67.4	96.3	86.2	42.0	130.3	36.91
	R101	60.8	92.0	68.0	96.6	88.0	74.7	248.2	49.74
CMDPNet	**(Ours)**	**62.6**	**92.8**	**71.4**	**96.9**	**88.4**	24.9	91.5	26.58
Gain vs. best (pp)		+1.1	+0.2	+3.0	+0.1	+0.4	—	—	—

Note: accuracy is the test metric for the seed-0 locked weights; complexity and latency follow the testing protocol in Sect. 4.6. Arrows mark the direction of improvement; bold values are column-wise best, gray marks the RT-DETRv1-R18 baseline. The rose gain row compares only the five detection-accuracy metrics, each against the best comparison model in that column, in percentage points (pp). Family and Variant are concatenated to form the model name; "—" for YOLOv7 denotes no suffix.

Equation (18) equally averages ten IoU thresholds and $C = 2$ classes (damage/drop); "Overall" likewise averages both classes.

$$\mathrm{mAP}_{50:95} = \frac{1}{10C} \sum_{c=1}^{C} \sum_{\tau \in \{0.50,\ldots,0.95\}} AP_{c,\tau}. \quad (18)$$

Group B selects structures by peak validation mAP50-95; SHSA is the best single module under seed 0. For each configuration/seed, weights are locked by fitness $= 0.1\,\mathrm{mAP}_{50} + 0.9\,\mathrm{mAP}_{50:95}$ (mAP50 denotes AP50) before frozen testing. Consequently, validation-peak and locked-weight epochs can differ.

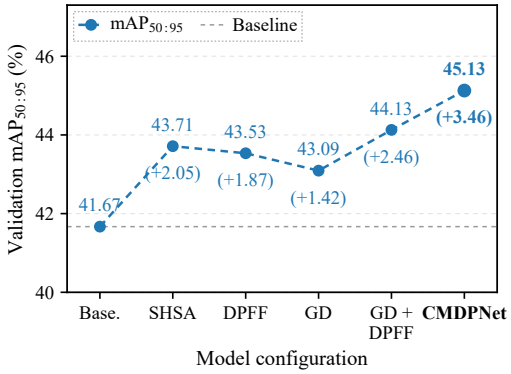
For seed-wise metrics x_i ($n = 3$), results report the mean and sample standard deviation (Eq. (19)).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \qquad s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}. \quad (19)$$

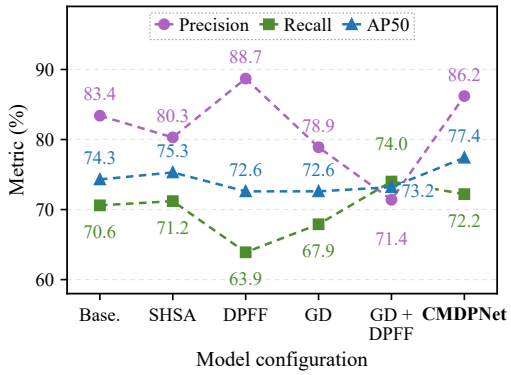
Table 4 Group B three-seed validation results

Metric ↑	Model		
	RT-DETR	SHSA	**CMDPNet**
mAP50-95	40.745 ± 1.544	43.217 ± 0.697	**44.594** ± 0.518
Δ (pp)	0.000	+2.472	**+3.849**
AP50	72.709 ± 3.525	73.226 ± 1.791	**76.718** ± 0.833

Note: values are mean ± sample standard deviation (n=3); accuracy is in percent. Δ is computed from the mean mAP50-95. RT-DETR and SHSA denote RT-DETR-R18 and the SHSA Encoder, respectively; bold marks the best mean, rose marks the gain over the baseline.



(a) Ablation accuracy and gain



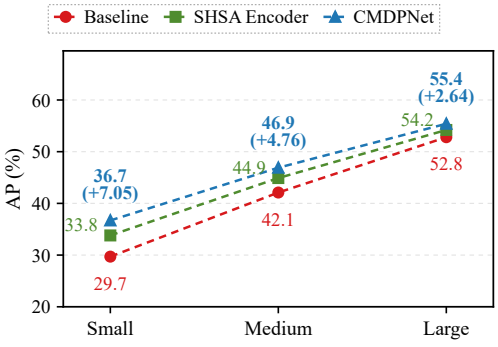
(b) Precision, recall and AP50

Fig. 6 Core structural ablation results. (a) Validation mAP50-95 for each configuration, with the gain over the baseline in parentheses (percentage points); (b) Precision, Recall, and AP50 at the corresponding epoch. The two panels keep two and one decimal place, respectively; Base. denotes Baseline.

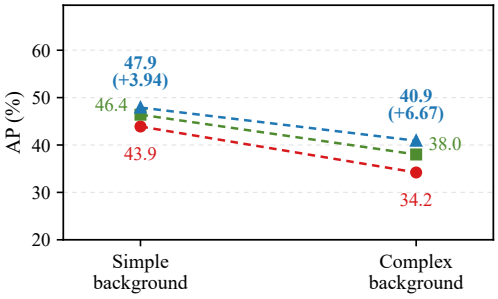
Table 5 Group B overall results on the frozen test set

Metric ↑	Model		
	RT-DETR	SHSA	**CMDPNet**
mAP50-95	39.132 ± 0.696	40.282 ± 2.171	**44.382** ± 0.602
Δ (pp)	0.000	+1.150	**+5.250**
AP50	70.555 ± 1.416	69.754 ± 2.500	**76.918** ± 1.136
AP75	39.161 ± 1.858	44.429 ± 1.120	**49.327** ± 1.282
Precision	84.430 ± 0.815	82.083 ± 5.738	**86.015** ± 1.104
Recall	65.950 ± 2.539	66.775 ± 2.799	**71.842** ± 1.026

Note: accuracy is in percent; Δ is the mean gain in mAP50-95 over the baseline. Model abbreviations and the meaning of bold/color follow Table 4. Values are mean ± sample standard deviation (n=3); the evaluated weights are the fitness-saved locked weights for each configuration and seed. P/R correspond to each model's best F1 operating point.



(a) Target scale



(b) Background complexity

Fig. 7 Target-scale and background-complexity analysis. (a) The three scales Small, Medium, and Large; (b) simple and complex backgrounds. The gain of CMDPNet over the baseline is given in parentheses (percentage points); AP and the gain are kept to one and two decimal places, respectively.

4.3 Comparison with Existing Detectors

Table 3 compares YOLO families [33–38] and RT-DETR variants [5, 23] under Group A. RT-DETR-R18 denotes RT-DETRv1-R18. CMDPNet leads all five metrics, improving mAP50-95/AP75 by 1.1/3.0 percentage points (pp) over the respective best comparisons.

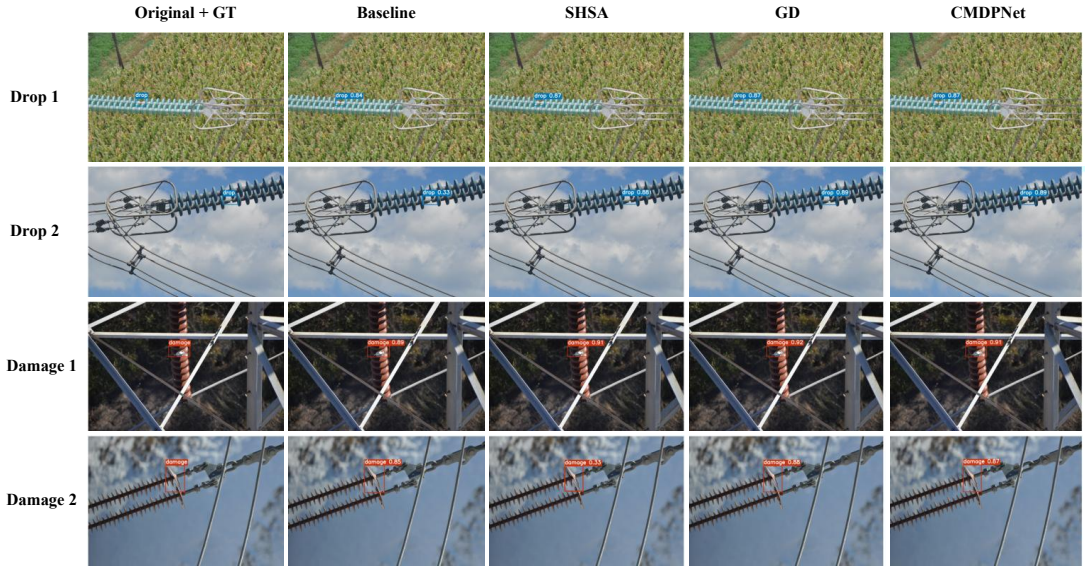


Fig. 8 Comparison of detection results on typical insulator defect samples. Columns are, in order, the ground truth and the predictions of the different detection structures; rows cover typical drop and damage scenes. Red boxes denote damage and blue boxes denote drop.

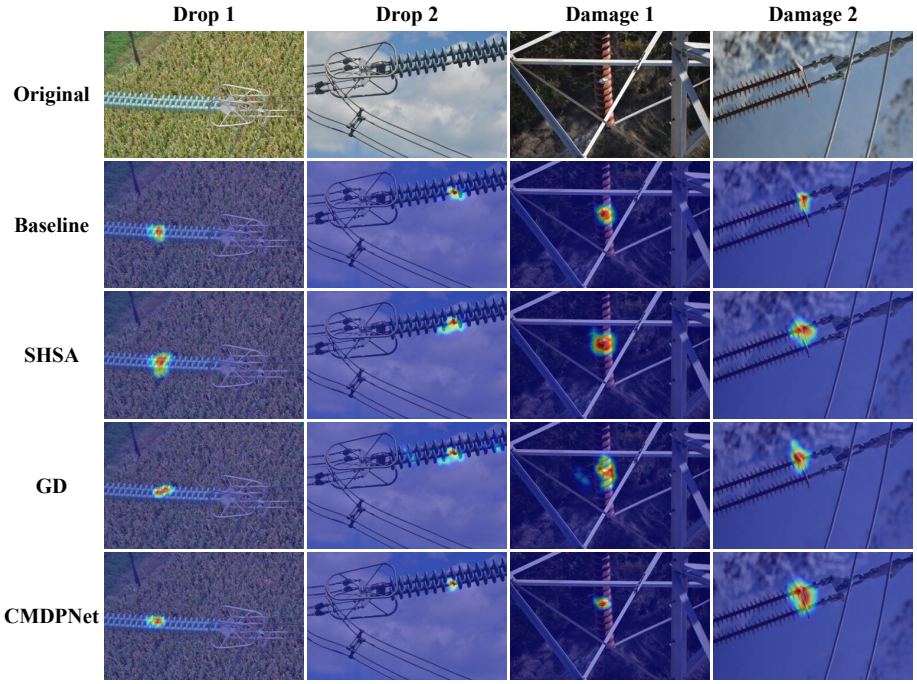


Fig. 9 Feature responses of different structures on typical insulator defect samples. All response maps are independently normalized and are used only to compare the location and extent of high-response regions, not to compare absolute activation strength across images.

4.4 Ablation and Multi-Seed Evaluation

On Group B seed-0 validation (Fig. 6), all modules improve mAP50-95; CMDPNet exceeds single modules at 45.128%, with 77.4% AP50. DPFF favors Precision; GD Neck + DPFF favors Recall.

GD Neck + DPFF reaches 44.128%, versus 41.669% for baseline (+2.459 pp) and 43.092/43.534% individually.

The interaction in Eq. (20) uses AP for mAP50-95, AP_0 for the baseline, and Δ_A, Δ_B, and Δ_{AB} for individual and combined gains.

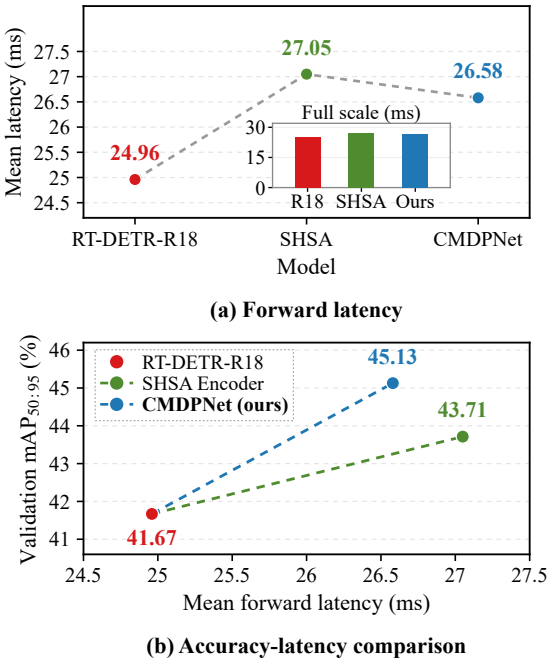


Fig. 10 Accuracy-latency comparison. (a) Average forward latency, with a zero-based full-scale inset; (b) pairing of Group B seed-0 validation mAP50-95 with the latency of the same structure. Dashed lines only aid comparison of the observed points.

$$\begin{aligned} I(A,B) &= \Delta_{AB} - \Delta_A - \Delta_B \\ &= AP(A+B) - AP(A) \\ &\quad - AP(B) + AP_0. \end{aligned} \tag{20}$$

The GD Neck/DPFF interaction is −0.829 pp, indicating subadditive gains on seed-0 validation.

Table 4 compares validation peaks and same-epoch AP50 across seeds 0–2 (200 epochs) for baseline, seed-0-selected SHSA, and CMDPNet. CMDPNet gains 3.849 pp in mean mAP50-95. Frozen testing (Table 5) yields 44.382% ± 0.602% mAP50-95 and mAP50-95/AP75 gains of 5.250/10.166 pp over RT-DETR-R18.

4.5 Per-Class and Qualitative Analysis

On the internal frozen test, CMDPNet's three-seed mean mAP50-95/AP75 is 30.637/30.178% for damage (gains 2.853/3.084 pp) and 58.127/68.476% for drop (gains 7.647/17.247 pp). Figure 7 explores target scale and background complexity.

Normalized box dimensions are mapped to 640 × 640 pixels. For area A, Small/Medium/Large correspond to $A < 32^2$, $32^2 \le A < 96^2$, and $A \ge 96^2$.

With clutter or weak boundaries, CMDPNet's boxes align more closely with defects (Fig. 8), with responses concentrated on defect locations (Fig. 9).

4.6 Model Complexity and Efficiency

Table 2 specifies the unified timing setup. Latency averages 3000 synchronized passes in PyTorch eval/inference_mode with random $1 \times 3 \times 640 \times 640$ inputs. Timing includes the detection head but excludes image reading, resizing, external filtering/NMS, and plotting. Seed-0 locked weights come from Group B for RT-DETR-R18/SHSA/CMDPNet and Group A for other detectors.

Figure 10 pairs Group B seed-0 validation-peak accuracy with locked-weight latency; the checkpoints may differ. RT-DETR-R18/SHSA/CMDPNet take 24.96/27.05/26.58 ms: CMDPNet adds 1.62 ms (6.5%) over baseline and saves 0.47 ms over SHSA. Their validation mAP50-95 values are 41.669/43.714/45.128%, giving CMDPNet gains of 3.459/1.414 pp. Its complexity is 24.9M parameters and 91.5 GFLOPs.

5 Conclusion

CMDPNet connects high-level context, cross-scale semantics, and P2 detail for four-scale insulator defect detection. Separate cross-detector and three-seed frozen tests support improved accuracy at a modest forward-latency cost. Evidence remains limited to an internal file-level TLID split and excludes complete deployment timing. Cross-line and cross-scene validation, alongside end-to-end edge-device testing including image processing, will assess the approach's broader applicability.

Declarations. **Author contributions** Linsihang Qiao performed all research activities, including conceptualization, methodology, software, experiments, data curation, analysis, visualization, and manuscript writing and revision.

Funding This study received no specific funding.

Competing interests The author declares no competing interests.

Data availability The TLID dataset used in this paper is from the published reference [17]; the 499/106/107 file-level split and training configuration used in this paper are available from the corresponding author upon reasonable request.

References

[1] Su, J., Yuan, Y., Przystupa, K., Kochan, O.: Insulator defect detection algorithm based on improved YOLOv8 for electric power. Signal, Image and Video Processing **18**(8-9), 6197–6209 (2024) https://doi.org/10.1007/s11760-024-03307-w

[2] Lan, X., Wang, X., Liu, L.: RMC-YOLO: An Object Detection Model for Insulator Defects. Signal, Image and Video Processing **19**(12), 1033 (2025) https://doi.org/10.1007/s11760-025-04568-9

[3] 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

[4] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Computer Vision – ECCV 2020, pp. 213–229 (2020). https://doi.org/10.1007/978-3-030-58452-8_13

[5] 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: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16965–16974 (2024). https://doi.org/10.1109/cvpr52733.2024.01605

[6] Yun, S., Ro, Y.: SHViT: Single-Head Vision Transformer with Memory Efficient Macro Design. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5756–5767 (2024). https://doi.org/10.1109/cvpr52733.2024.00550

[7] Wang, C., He, W., Nie, Y., Guo, J., Liu, C., Wang, Y., Han, K.: Gold-YOLO: Efficient Object Detector via Gather-and-Distribute Mechanism. In: Advances in Neural Information Processing Systems, vol. 36, pp. 51094–51112 (2023). https://doi.org/10.52202/075280-2224

[8] Liu, Y., Liu, D., Huang, X., Li, C.: Insulator defect detection with deep learning: A survey. IET Generation, Transmission & Distribution **17**(16), 3541–3558 (2023) https://doi.org/10.1049/gtd2.12916

[9] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Advances in Neural Information Processing Systems, vol. 28, pp. 91–99 (2015). https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[10] Wen, Q., Luo, Z., Chen, R., Yang, Y., Li, G.: Deep Learning Approaches on Defect Detection in High Resolution Aerial Images of Insulators. Sensors **21**(4), 1033 (2021) https://doi.org/10.3390/s21041033

[11] Zhang, T., Zhang, Y., Xin, M., Liao, J., Xie, Q.: A Light-Weight Network for Small Insulator and Defect Detection Using UAV Imaging Based on Improved YOLOv5. Sensors **23**(11), 5249 (2023) https://doi.org/10.3390/s23115249

[12] Yu, Z., Lei, Y., Shen, F., Zhou, S., Yuan, Y.: Research on Identification and Detection of Transmission Line Insulator Defects Based on a Lightweight YOLOv5 Network. Remote Sensing **15**(18), 4552 (2023) https://doi.org/10.3390/rs15184552

[13] Li, D., Lu, Y., Gao, Q., Li, X., Yu, X., Song, Y.: LiteYOLO-ID: A Lightweight Object Detection Network for Insulator Defect Detection. IEEE Transactions on Instrumentation and Measurement **73**, 1–12 (2024) https://doi.org/10.1109/tim.2024.3418082

[14] Zhang, L., Li, B., Cui, Y., Lai, Y., Gao, J.: Research on improved YOLOv8 algorithm for insulator defect detection. Journal of Real-Time Image Processing **21**(1), 22 (2024) https://doi.org/10.1007/s11554-023-01401-9

[15] Wang, P., Liu, Z., Cheng, X., Tang, M., Li, Y.: Study on defect detection in lightweight insulators based on improved YOLOv8. Journal of Computational Methods in Sciences and Engineering **24**(6), 3993–4003 (2024) https://doi.org/10.1177/14727978241304269

[16] He, Z., Yang, W., Liu, Y., Zheng, A., Liu, J., Lou, T., Zhang, J.: Insulator Defect Detection Based on YOLOv8s-SwinT. Information **15**(4), 206 (2024) https://doi.org/10.3390/info15040206

[17] Hu, Z., Zhai, Y., Zhao, Z., Wang, Q., Zhai, B., Yang, K., Hu, P.: Towards Defect Detection of Transmission Line Insulator: A Dataset, Benchmarks and Challenges. In: 2025 10th International Conference on Power and Renewable Energy (ICPRE), pp. 473–482 (2025). https://doi.org/10.1109/icpre67300.2025.11274094

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[19] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=YicbFdNTTy

[20] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9992–10002 (2021). https://doi.org/10.1109/iccv48922.2021.00986

[21] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=gZ9hCDWe6ke

[22] Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.-Y.: DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In: International Conference on Learning Representations (2023). https://openreview.net/forum?id=3mRwyG5one

[23] Wang, S., Xia, C., Lv, F., Shi, Y.: RT-DETRv3: Real-Time End-to-End Object Detection with Hierarchical Dense Positive Supervision. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1628–1636 (2025). https://doi.org/10.1109/wacv61041.2025.00166

[24] 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

[25] 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

[26] 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

[27] Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: Convolutional Block Attention Module. In: Computer Vision – ECCV 2018. Lecture Notes in Computer Science, vol. 11211, pp. 3–19 (2018). https://doi.org/10.1007/978-3-030-01234-2_1

[28] 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

[29] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=Bkg6RiCqY7

[30] Zhang, H., Wang, Y., Dayoub, F., Sünderhauf, N.: VarifocalNet: An IoU-aware Dense Object Detector. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8510–8519 (2021). https://doi.org/10.1109/cvpr46437.2021.00841

[31] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 658–666 (2019). https://doi.org/10.1109/cvpr.2019.00075

[32] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Computer Vision – ECCV 2014, pp. 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48

[33] Li, C., Li, L., Geng, Y., Jiang, H., Cheng, M., Zhang, B., Ke, Z., Xu, X., Chu, X.: YOLOv6 v3.0: A Full-Scale Reloading. arXiv preprint arXiv:2301.05586 (2023) https://doi.org/10.48550/arXiv.2301.05586

[34] Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7464–7475 (2023). https://doi.org/10.1109/cvpr52729.2023.00721

[35] 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. Lecture Notes in Computer Science, vol. 15089, pp. 1–21 (2024). https://doi.org/10.1007/978-3-031-72751-1_1

[36] 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

[37] Xu, X., Jiang, Y., Chen, W., Huang, Y., Zhang, Y., Sun, X.: DAMO-YOLO : A Report on Real-Time Object Detection Design. arXiv preprint arXiv:2211.15444 (2022) https://doi.org/10.48550/arXiv.2211.15444

[38] Chen, Y., Yuan, X., Wang, J., Wu, R., Li, X., Hou, Q., Cheng, M.-M.: YOLO-MS: Rethinking Multi-Scale Representation Learning for Real-Time Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(6), 4240–4252 (2025) https://doi.org/10.1109/tpami.2025.3538473