applied sciences

MDPI

Article

DRB-DEIM: A Text-Guided, Orientation-Aware Multiscale Detector for Transmission-Line Insulator Defects

Huijia Liu, Heng Wu * and Rui Jia

College of Electrical Engineering and New Energy, China Three Gorges University, Yichang 443002, China; 18872821177@163.com (H.L.); 19571461494@163.com (R.J.)
* Correspondence: faijfa2048@gmail.com

Featured Application

DRB-DEIM can be deployed on the NVIDIA Jetson Nano to detect and localize insulator breakage and flashover traces in UAV-acquired transmission-line inspection images. It supports local image processing for automated defect screening and inspection review.

Abstract

To address the loss of local defect information, insufficient adaptation to irregular defect patterns, and background interference in transmission-line insulator inspection, this paper proposes DRB-DEIM, a text-guided and orientation-aware multiscale detector based on DEIM-D-FINE-S. The model uses offline text prototypes to provide class-level semantic guidance and introduces three visual components: Detail-Preserving Stem for preserving local information during early downsampling, Orientation-Routed Inception ELAN for adapting feature extraction to directional and irregular defect patterns, and Bidirectional Routing-Guided Feature Pyramid Network for content-adaptive multiscale feature fusion. Experiments on an insulator defect dataset constructed from public datasets and field inspection images show that DRB-DEIM achieves 89.9% precision, 83.5% recall, 88.1% mAP@50, and 59.2% mAP@50–95. Compared with DEIM-D-FINE-S, mAP@50 and mAP@50–95 are improved by 5.3 and 8.9 percentage points, respectively. The proposed model contains 11.57 M parameters and reaches 117.4 frames per second on the workstation test platform. It also achieves an average mAP@50 of 71.7% under six degraded imaging conditions. Deployment on a Jetson Nano achieves 87.3% mAP@50 and 20.83 frames per second using TensorRT FP16. These results demonstrate improved detection of small and weakly visible insulator defects and support the feasibility of local inference on a resource-constrained edge platform.

Keywords: transmission-line insulator; defect detection; text-guided detection; multi-scale feature fusion; real-time detection; UAV inspection

Check for updates

Academic Editor: Pedro Couto

Received: 26 September 2026
Revised: 12 October 2026
Accepted: 13 October 2026
Published: 17 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Insulators provide electrical isolation and mechanical support in transmission lines, and their condition directly affects the safety and stability of power systems. Prolonged electrical stress, mechanical loading, wind-blown sand, rain, snow, and surface contamination can cause shed breakage, material loss, flashover, and surface ablation. If such defects are not identified promptly, insulation performance may deteriorate and, in severe cases, lead to line trips or power interruptions. Because transmission lines extend over long distances and often traverse difficult terrain, manual inspection is labor-intensive, subjective, and

may fail to detect defects. With the development of Unmanned Aerial Vehicles (UAVs) and high-resolution imaging systems, automated inspection using aerial images has gradually been applied to improve the efficiency of transmission-line maintenance [1–5].

With the development of deep learning, most work on insulator defect inspection now uses end-to-end visual recognition. Two-stage detectors, such as the Faster Region-based Convolutional Neural Network (Faster R-CNN), provide effective feature extraction and object localization. However, their region proposal and multistage prediction processes increase inference costs [6]. You Only Look Once (YOLO) offers fast inference through a single-stage architecture and is widely used for UAV-based power-line inspection. To solve the problems of scale variation and background interference, previous studies have added small-object detection heads, attention mechanisms, improved Feature Pyramid Networks (FPNs), and lightweight convolutional blocks [1–5,7,8]. Although these designs improve the representation of whole insulators and noticeable damage, they are still based on appearance. Defects can be difficult to recognize when they are small or have weak local contrast. Nearby specular reflections, conductor edges, or bright background regions can also obscure defect cues. Under these conditions, visual appearance alone may not provide reliable class discrimination.

Objects in transmission-line images vary in scale and shape. A complete insulator usually occupies a larger image area and has a continuous, regularly spaced arrangement of sheds. Broken defects include localized gaps, fractured edges, and missing material. Flashover defects often appear as small, irregular discharge traces or low-contrast ablated regions. Repeated spatial downsampling can weaken edge and texture cues in local defects. Fixed-shape kernels may respond inconsistently as insulator pose and defect direction change. Conventional feature pyramids use fixed alignment and fusion rules, which may introduce background noise or obscure shallow localization cues during cross-level propagation. Therefore, the main problems in insulator defect detection are to retain local defect information, adapt to orientation-dependent anomalies, and improve multiscale fusion without introducing an overly complex model.

Transformer-based detectors combine global relationship modeling with end-to-end prediction. Detection Transformer (DETR) formulates object detection as set prediction and uses bipartite matching to assign predictions to ground-truth objects during training [9]. Deformable DETR uses sparse multi-scale sampling to reduce attention cost and improve scale-aware representation [10]. Subsequent methods have further improved query initialization, denoising training and efficient hybrid encoding in DETR with Improved DeNoising Anchor Boxes (DINO), the Real-Time Detection Transformer (RT-DETR) and RT-DETRv2 [11–13]. The Fine-Grained Distribution Refinement model (D-FINE) reformulates box regression as an iterative refinement process and introduces localization self-distillation to improve supervision consistency among decoder layers [14]. DETR with Improved Matching (DEIM) increases the density of positive matches and modifies classification supervision based on prediction-target matchability to accelerate convergence and stabilize query assignment [15]. Together, DEIM and D-FINE offer end-to-end prediction and precise localization; thus, DEIM-D-FINE can be applied to address the problem of local insulator defects.

Vision–language pre-training has also created an effective way to introduce category semantics for visual detectors. Contrastive Language–Image Pretraining (CLIP) learns a shared representation space from large-scale image–text pairs [16], whereas Text–Image Pretraining with Spatial Awareness v2 (TIPSv2) strengthens fine-grained alignment between local image regions and textual descriptions [17]. Vision and Language Knowledge Distillation (ViLD), Region-Based Language–Image Pretraining (RegionCLIP), Grounded Language–Image Pretraining (GLIP), and Grounding DINO demonstrate the value of align-

ing region-level visual representations with language semantics for open-vocabulary detection [18–21]. YOLO-World introduces visual–textual interaction and region–text contrastive learning into efficient YOLO-based detection [22], while YOLOE uses reparameterizable region–text alignment to support efficient text-prompted detection [23]. Vision–language information has also been used in power-line and industrial defect inspection to supplement category attributes and domain knowledge that are difficult to encode through visual appearance alone [4,24,25]. These findings suggest that natural-language descriptions of observable class properties can provide stable semantic references for fine-grained defect discrimination.

Simply attaching text semantics to a detector, however, does not resolve the recognition of localized insulator defects. A text prototype must be associated with visual evidence that retains explicit spatial and morphological information. If broken edges or flashover textures have already been weakened during shallow downsampling, feature transformation, or cross-scale fusion, subsequent semantic matching cannot recover their locations reliably. Likewise, text information confined to the final classifier has limited influence on visual feature formation and cannot consistently guide candidate selection or query refinement. Local-detail preservation, orientation-aware morphology modeling, cross-scale aggregation, and text–visual matching should therefore be integrated within a single detection pipeline. This allows category semantics to guide feature calibration, query generation, and final classification.

To address these issues, this study proposes Detail-Preserving, Routing-Aware, and Bidirectional-Fusion DEIM (DRB-DEIM), a text-guided and orientation-aware detector for transmission-line insulator defects. Built on DEIM-D-FINE-S, the model focuses on preserving and reorganizing local defect evidence before matching it to category semantics. The Detail-Preserving Stem (DPStem) combines progressive downsampling with a direct pixel-rearrangement path, allowing fine image information to reach the shallow feature representation through complementary routes. The Orientation-Routed Inception Efficient Layer Aggregation Network (ORI-ELAN) introduces input-dependent weighting of directional branches within the existing Efficient Layer Aggregation Network (ELAN) topology and uses local contrast to represent discontinuities in repetitive shed structures. The Bidirectional Routing-Guided Feature Pyramid Network (BRG-FPN) combines image-level scale preferences with spatially varying fusion weights while retaining the target-level feature through a residual path. A frozen TIPSv2 text encoder generates offline class prototypes, which provide a common semantic reference for feature calibration, query selection, and decoder classification. These changes address information loss at successive visual stages and connect the resulting defect representations to consistent category guidance. The proposed model is further evaluated on an NVIDIA Jetson Nano B01 4 GB platform using PyTorch 1.10 and TensorRT 8.2.1 to examine detection accuracy, model inference efficiency, and board-level power consumption.

2. DEIM-D-FINE Model

DEIM-D-FINE-S uses D-FINE as the detector and the DEIM training strategy to improve matching efficiency in end-to-end detection. The architecture comprises a High-Performance GPU Network V2 (HGNetV2) backbone, a hybrid encoder, and a D-FINE Transformer decoder. The backbone extracts three hierarchical feature maps, P3, P4, and P5. The hybrid encoder projects the three feature maps into a shared embedding space and models global context on the lowest-resolution feature map, P5. Cross-scale fusion then combines spatial information with high-level semantics. The candidate queries are taken from the encoded features and passed to the decoder, and multi-scale deformable attention updates them layer by layer. D-FINE replaces direct coordinate regression with iterative

refinement of boundary probability distributions via Fine-Grained Distribution Refinement (FDR), and its Global Optimal Localization Self-Distillation (GO-LSD) strengthens localization supervision across decoder layers. During training, DEIM uses dense one-to-one (O2O) matching to increase the number of valid positives and employs Matchability-Aware Loss (MAL) to supervise the matches based on their quality. The above mechanisms reduce the sparsity of supervision in traditional O2O matching and improve both convergence and localization [14,15].

Despite its favorable accuracy and training efficiency, DEIM-D-FINE-S learns category representations primarily from visual supervision and lacks an explicit semantic reference for equipment condition and defect attributes. In inspection images, insulator denotes the complete device, whereas broken and flashover denote localized structural damage and surface discharge traces. The three classes differ in semantic granularity and spatial extent and may coexist through containment or adjacency. When class boundaries are learned only from appearance, the repetitive shed pattern of the complete insulator may dominate feature aggregation and suppress responses from small broken edges or faint flashover traces. Specular fittings, local highlights, and cluttered textures can also resemble flashover. Category-specific text prototypes are therefore introduced as stable semantic anchors for feature calibration, query selection, and class discrimination.

The visual feature formation process also remains imperfect for localized defect representation. First, strided convolution at the input compresses spatial resolution before informative features are fully established, weakening high-frequency evidence such as gaps, fracture boundaries, and faint flashover traces. Second, RepNCSPELAN4 offers efficient feature reuse through channel splitting, serial reparameterized cross-stage partial (RepCSP) transformations, and terminal aggregation, but its internal paths rely mainly on fixed convolutional kernels and do not explicitly adjust the balance of directional responses according to the input context. Third, the original feature pyramid uses a fixed-scale transformation and fusion path, does not explicitly weigh the contribution of different levels according to the content of the image, and thus may introduce background texture and attenuate small anomalies. Therefore, DRB-DEIM rebuilds the visual path in three steps: shallow detail preservation, orientation-aware morphology modeling, and adaptive cross-scale fusion, and integrates these stages with text-guided calibration and classification.

3. DRB-DEIM

DRB-DEIM is built on DEIM-D-FINE-S and comprises a visual feature extraction branch, an offline text-prototype branch, and a Text-Guided D-FINE (TG-D-FINE) decoder, as illustrated in Figure 1.

The model retains DEIM dense matching and D-FINE fine-grained box refinement. Its visual redesign addresses the successive stages at which localized defect information can be weakened. DPStem carries fine image information into the backbone, ORI-ELAN adjusts the balance of directional responses during feature transformation, and BRG-FPN controls how these responses are combined across resolutions. The cached text prototypes then provide shared category references for the resulting features and detection queries. This arrangement allows semantic matching to operate on visual representations that retain local structure and multiscale context.

In the visual branch, DPStem first downsamples the input by a factor of four through complementary convolutional, smooth-sampling, and pixel-rearrangement paths. HGNetV2 then extracts hierarchical features P_3, P_4, and P_5. ORI-ELAN replaces the transformation path in the original RepNCSPELAN4 block and reorganizes features through orientation routing and multidirectional local modeling. BRG-FPN subsequently performs bidirectional cross-scale aggregation and adjusts the contribution of each level according to

its content, producing the multiscale outputs O_3, O_4, and O_5 with complementary spatial detail and semantic context.

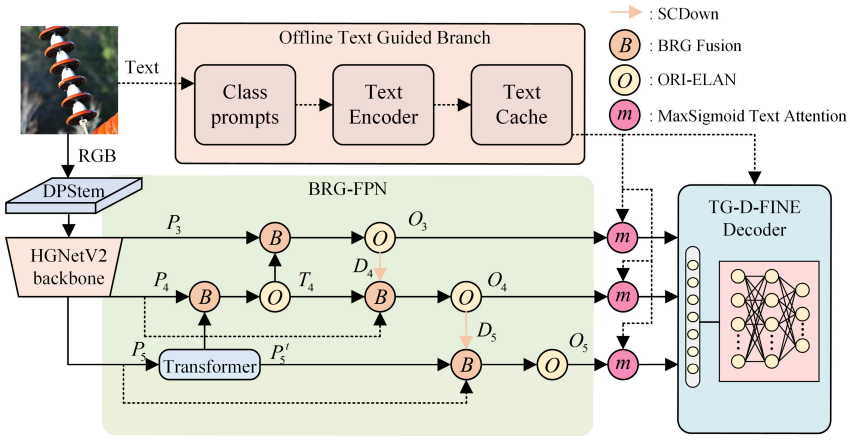


Figure 1. Overall architecture of the Detail-Preserving, Routing-Aware, and Bidirectional-Fusion DEIM (DRB-DEIM). TG denotes text-guided.

Before detector training, class prompts are encoded by the frozen TIPSv2 text encoder. The normalized text embeddings are stored in an offline cache and loaded once for both training and inference. After dimensional alignment by a text adapter, the prototypes participate in MaxSigmoid text attention at all three feature scales and emphasize local regions that are semantically related to a target class. The calibrated features are used for encoder candidate selection, while the adapted text prototypes serve as classification references in the TG-D-FINE decoder. The same prototypes thus provide a common semantic reference for query selection and decoder classification.

This architecture connects shallow-detail preservation, orientation-aware modeling, bidirectional multiscale fusion, and category-semantic matching within one detector. The visual branch retains spatial and textural evidence for local defects, while the cached prototypes provide consistent semantic guidance throughout detection. Because text features are generated in advance, the text encoder is absent from the online inference path so the detector maintains high throughput.

3.1. Text-Guided Branch

An offline text-guided branch is constructed to introduce class semantics without incurring online text-encoding cost. The text encoder from the TIPSv2 vision–language pre-trained model generates prototypes from prompts designed for the observable properties of the classes in transmission-line inspection imagery. All prompts describe the inspection setting and target type. Class-specific descriptions cover the overall insulator structure, localized structural damage, and flashover traces. The process is shown in Figure 2.

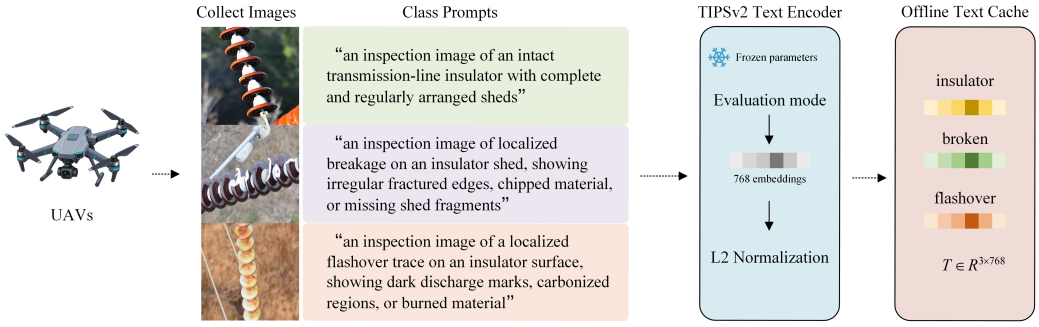


Figure 2. Class prompts and generation of the offline text-feature cache.

Each prompt is tokenized with SentencePiece and passed through the frozen TIPSv2 text encoder. The resulting class embeddings are L2-normalized to unit-length prototypes, and cached with their class indices in a fixed order. Training and inference load these cached features directly. These prototypes provide class references derived from vision–language pretraining and defect descriptions. Randomly initialized learnable class embeddings instead acquire their representations from the target detection dataset.

For existing classes, revised prompts can be encoded offline to update the cache without changing the detector weights. Adding a new defect class requires updating the cached prototypes and class-index mapping, followed by supervised fine-tuning on annotated images covering the expanded class set. Fine-tuning can start from the existing detector weights rather than training from scratch, while the TIPSv2 text encoder remains frozen.

The cached 768-dimensional prototypes are passed through a lightweight adapter composed of Root Mean Square Layer Normalization (RMSNorm), a gated residual feed-forward layer, and a linear projection. The adapter is trained with the detector to map the pretrained class references into the 256-dimensional visual space. The adapted prototypes are then applied to the three BRG-FPN outputs through MaxSigmoid text attention.

The attention gates and detection scores are computed from the adapted prototypes and current visual features. Their responses therefore vary with the input image, allowing the same class references to guide different spatial regions.

Let $\widetilde{N}_l$ denote the visual feature at scale l, $e_l(u,v)$ the visual embedding at position (u,v), and t_c the adapted and projected prototype of class C. Text attention at scale l is defined as:

$$a_l(u,v) = \sigma\left[\max_C\left(\frac{e_l(u,v)^{\mathrm{T}} t_c}{\sqrt{d}}\right) + b_l\right], \quad \widetilde{N}_l = P_l(N_l) \odot a_l \tag{1}$$

where d is the shared embedding dimension, b_l is a learnable bias, $P_l(\cdot)$ is a lightweight visual projection, and $\odot$ denotes element-wise multiplication. At each spatial position, the largest class-prototype similarity is converted into a scalar gate that recalibrates the projected visual feature. This formulation is well suited to the three-class detection task because it maintains the spatial layout of the multiscale features while directly measuring their correspondence with the cached class prototypes. Cross-attention emphasizes token-to-token interaction, whereas feature-wise linear modulation (FiLM) performs channel-wise affine modulation. MaxSigmoid attention instead provides position-wise semantic calibration with a lightweight gating operation and uses the same prototype-based similarity relation as candidate-query selection and decoder classification.

The maximum operation is differentiable whenever the largest class similarity is unique. To examine its gradient, spatial and scale indices are omitted. Let s_c denote the unscaled visual–text similarity for class c. The attention gate is written as

$$a = \sigma(z), \quad z = \frac{\max_c s_c}{\sqrt{d}} + b \tag{2}$$

For a unique maximizing class c^*, the derivative is

$$\frac{\partial a}{\partial s_c} = \frac{a(1-a)}{\sqrt{d}} \mathbf{1}_{c=c^*}, \quad \left|\frac{\partial a}{\partial s_c}\right| \leq \frac{1}{4\sqrt{d}} \tag{3}$$

1 denotes the indicator function. At tied maxima, the maximum admits subgradients supported on the maximizing classes. The implementation propagates the gradient through the first maximizing entry, selecting a valid subgradient of the maximum. Since the sigmoid derivative is bounded by $1/4$, the same bound holds for the selected backward gradient. Thus, the gate has a bounded backward sensitivity to its class-similarity inputs.

The calibrated multiscale features are subsequently passed to the TG-D-FINE decoder. During candidate-query generation, cosine similarity between encoder features and class prototypes measures the category relevance of each spatial location, and high-confidence locations initialize the detection queries. At every decoder layer, the classification head evaluates query features against the same prototypes in the shared embedding space. Candidate generation and classification at each decoder layer therefore share the same class references. Only the lightweight text adapter and similarity computation are used in the online path; the text encoder is not run again.

3.2. Detail-Preserving Stem

A conventional stem rapidly reduces the input resolution through successive strided convolutions. Although computationally efficient, this operation compresses the image before shallow representations are fully formed, which may attenuate small gaps, broken boundaries, and weak flashover textures. The proposed Detail-Preserving Stem (DPStem), shown in Figure 3, instead performs two progressive downsampling stages to reach one-quarter resolution and establishes complementary paths for spatial modeling, low-aliasing sampling, and pixel rearrangement.

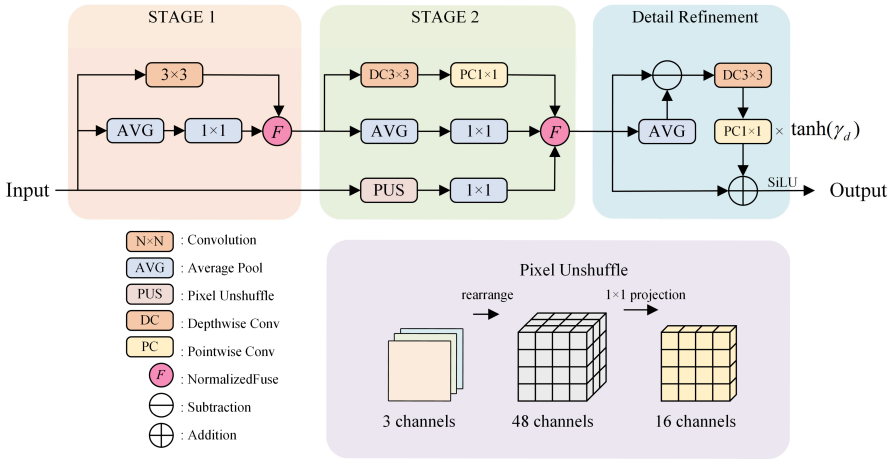


Figure 3. Architecture of the Detail-Preserving Stem (DPStem).

At the first stage, the image is processed in parallel by a convolution branch and an average-pooling branch. The former captures local edges and textures, whereas the latter smooths the input before subsampling and reduces high-frequency aliasing. Their equal-resolution outputs are fused with normalized learnable weights. At the second stage, the fused feature enters a depthwise-separable convolution branch and another average-pooling projection branch. In parallel, the original image is rearranged directly to the target resolution by Pixel Unshuffle with a scale factor of four. For a red–green–blue (RGB) input, each 4×4 neighborhood is moved to the channel dimension, expanding three channels to 48 before a projection. Because rearrangement does not discard sampled values, local color changes, edge discontinuities, and ablation cues can be delivered directly to the backbone.

To prevent fixed summation from obscuring useful branch responses, DPStem uses positive normalized fusion. For K paths at a given stage, let $B_k(\cdot)$ denote the transformation of the kth path. Their fusion is defined as:

$$\mathcal{F}(B_1, \ldots, B_K) = \sum_{k=1}^{K} \bar{\alpha}_k B_k, \quad \bar{\alpha}_k = \frac{\mathrm{ReLU}(\alpha_k)}{\varepsilon + \sum_{j=1}^{K} \mathrm{ReLU}(\alpha_j)} \tag{4}$$

where α_k is the learnable parameter for the kth path, its normalized value controls the contribution of that path, and ε ensures numerical stability. The rectified linear unit (ReLU) keeps the fusion weights nonnegative. K equals two and three in the first and second stages, respectively. The network can therefore balance convolutional modeling, smooth sampling, and pixel rearrangement according to the training data without introducing a costly spatial-attention module.

After the two fusion stages, a lightweight branch extracts high-frequency detail by subtracting the local average from the fused feature. Depthwise and pointwise convolutions then refine this component. The final output is written as:

$$Y_{\mathrm{DP}} = \mathrm{SiLU}\big(X_2 + \tanh(\gamma_d)\mathrm{PW}\big(\mathrm{DW}_{3\times3}\big(X_2 - \mathrm{AvgPool}_3(X_2)\big)\big)\big) \tag{5}$$

where $\mathrm{AvgPool}_3(\cdot)$ denotes 3×3 average pooling, and $\mathrm{DW}(\cdot)$ and $\mathrm{PW}(\cdot)$ denote depthwise and pointwise convolution, respectively. γ_d is a channel-wise learnable scale, and $\odot$ denotes element-wise multiplication. γ_d is initialized to zero so that the detail-refinement branch does not alter the fused feature at the beginning of training. During training, the detail-refinement branch learns to refine high-frequency cues associated with gaps, fractured boundaries, and ablation textures.

The direct pixel-rearrangement path bypasses the first downsampling stage and delivers original image samples to the second-stage fusion. Together with progressive downsampling and learnable detail refinement, it preserves local color and edge cues for subsequent feature extraction.

3.3. Orientation-Routed Inception ELAN

RepNCSPELAN4 reuses features through channel splitting, serial RepCSP transformations, and final aggregation. Its transformation path, however, is mainly built from fixed-shape convolution kernels. This limits its ability to handle UAV images in which insulator strings may appear horizontally, vertically, or obliquely, while broken edges and flashover marks extend in different directions. A kernel that responds well to one orientation may produce a weaker response when the target pose changes. The proposed Orientation-Routed Inception Efficient Layer Aggregation Network (ORI-ELAN) therefore replaces the original transformation path while retaining the split-and-aggregate structure of ELAN.

As shown in Figure 4, the input is first projected by a 1×1 convolution and divided into a shortcut branch and a transformation branch. The shortcut passes shallow information directly to the aggregation stage. The transformation branch contains two consecutive stages, each formed by an Orientation-Routed Inception (ORI) block followed by a convolution. Their outputs are concatenated with the shortcut feature and compressed by a final projection. In this way, orientation modeling is introduced inside the original ELAN topology without adding extra network stages or interrupting its short gradient paths.

Within each ORI block, only part of the input channels is processed by directional operators. These channels are divided among an identity branch and three different depthwise-convolution branches. The two strip-shaped kernels collect information along horizontal and vertical structures, whereas the square kernel focuses on local gaps and irregular surface damage. The identity branch preserves channels that do not require directional transformation. Processing only a subset of channels enlarges the available receptive-field shapes without applying all four operations to the entire feature map.

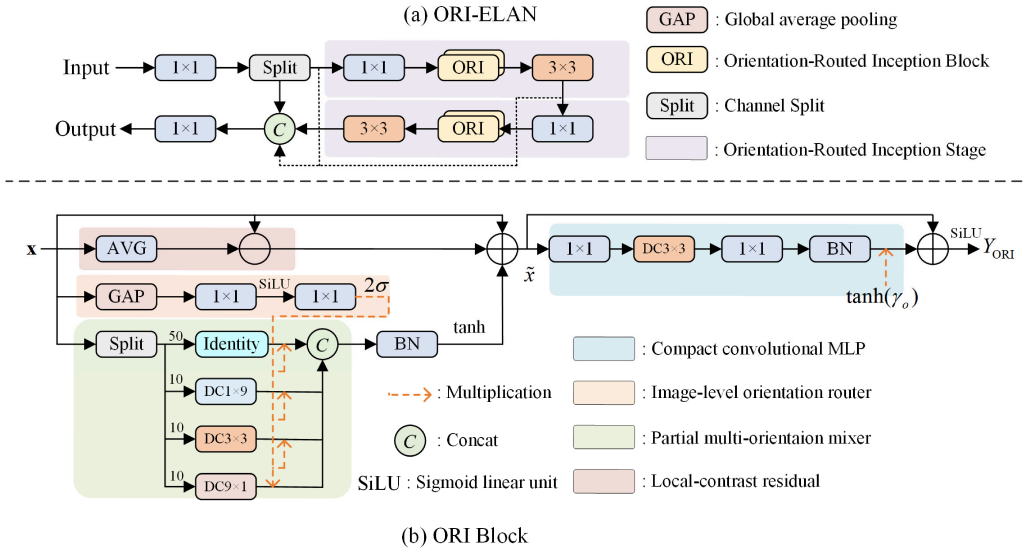


Figure 4. The Orientation-Routed Inception Efficient Layer Aggregation Network (ORI-ELAN) and the Orientation-Routed Inception (ORI) block. GAP, global average pooling; AVG, average pooling; DC, depthwise convolution; BN, batch normalization; SiLU, sigmoid linear unit; MLP, multilayer perceptron.

A fixed combination of these branches does not explicitly adjust their relative contributions to the image context. ORI therefore uses an image-level router to adjust their contributions. The router summarizes the input by global average pooling (GAP) and predicts four coefficients through two 1×1 mappings:

$$r = 2\sigma(W_2\delta(W_1\mathrm{GAP}(X))), \quad r = [r_0, r_h, r_s, r_v] \tag{6}$$

where X is the input feature; W_1 and W_2 are the two 1×1 mappings; $\delta(\cdot)$ denotes the sigmoid linear unit (SiLU); $\sigma(\cdot)$ is the sigmoid function; and r contains the four routing coefficients. The factor 2 constrains the coefficients to (0, 2). The resulting soft weights keep the identity, horizontal, vertical, and local branches active, allowing their responses to be combined according to the image context.

At each forward pass, each ORI block computes four routing coefficients for every image from its current input features. The router parameters remain fixed during inference, while the coefficients depend on the input. Values above 1 enhance a branch response, while values below 1 attenuate it. Each coefficient is shared across spatial locations within its block. The weighted branch outputs are concatenated, normalized, and added to the input through a residual connection. Routing adjusts the image-level balance among the branches, while their feature responses remain spatially varying. The local 3×3 branch and local-contrast residual provide complementary features for small, irregular defects, including those whose orientation differs from that of the insulator string. This design combines global contextual modulation with local feature extraction.

ORI also includes a local-contrast residual to recover discontinuities within the repetitive shed structure. A 3×3 local average is subtracted from the input, highlighting broken boundaries and abrupt intensity changes that may be weakened by surrounding intact sheds. The fused feature is subsequently refined by a compact convolutional multilayer perceptron (MLP), consisting of 1×1 channel expansion, 3×3 depthwise convolution, and 1×1 channel compression. Learnable channel-wise scales control the residual updates from the directional, local-contrast, and convolutional-MLP paths so that these responses can be introduced gradually during training.

3.4. Bidirectional Routing-Guided Feature Pyramid Network

The baseline feature pyramid combines aligned features through a fixed fusion rule, although complete insulators and localized defects require different spatial detail. BRG-FPN introduces content-dependent fusion at each node through complementary global and local routing. The global route estimates the relative contribution of each scale from the image content, while the local route adjusts that contribution at individual spatial positions. Their combined scores determine the fusion weights across scales. A separate target-level residual preserves the original feature while cross-scale information is introduced through a learnable gate. Within the bidirectional pyramid, this design allows fusion to vary between defect regions and their surroundings while maintaining the spatial information of the receiving level.

As illustrated in Figure 1, BRG-FPN receives P_3 and P_4 from the backbone together with Transformer-encoded P_5. Its top-down path transfers deep semantics toward high-resolution levels to strengthen category representation for local defects. The bottom-up path uses spatial–channel decoupled downsampling (SCDown) to return shallow localization cues to deeper levels. ORI-ELAN follows each fusion node and further consolidates orientation structure and local morphology in the aggregated feature.

Cross-scale interaction is implemented by Bidirectional Routing-Guided Fusion (BRG-Fusion), as shown in Figure 5. Each node accepts two or three inputs. A 1×1 convolution first projects every input to 256 channels. Higher-resolution features are downsampled by adaptive average pooling, lower-resolution features are upsampled by bilinear interpolation, and a feature already at the target resolution follows an identity path. All inputs thus share the same channel width and spatial size before adaptive fusion.

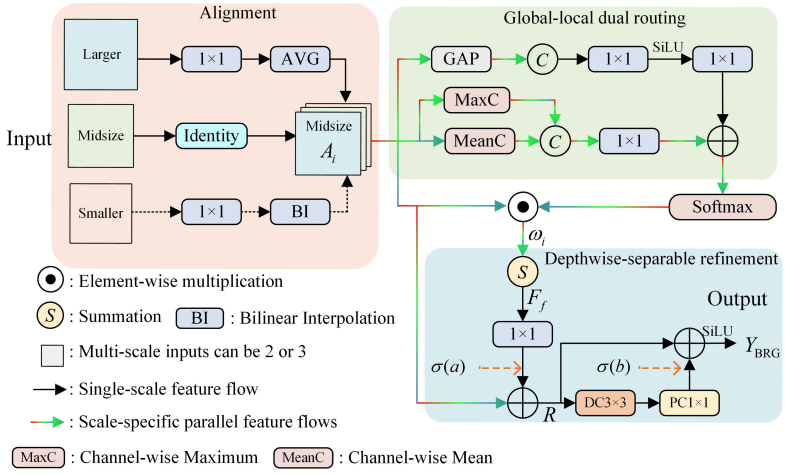


Figure 5. Structure of the Bidirectional Routing-Guided Fusion (BRGFusion).

After alignment, BRGFusion assesses each input at both image and spatial levels. The global path applies global average pooling to each scale, concatenates the descriptors, and predicts scale scores with two 1×1 mappings. The local path computes channel-wise means and maxima and produces location-dependent response maps through a lightweight projection. The global and local scores are added and normalized across scales:

$$\omega_i(u,v) = \frac{\exp(g_i + l_i(u,v))}{\sum_{j=1}^{K} \exp(g_j + l_j(u,v))} \tag{7}$$

where g_i is the image-level score of the ith input scale, $l_i(u,v)$ is its local score at position (u,v), K is the number of inputs to the current fusion node, and $\omega_i(u,v)$ is the normal-

ized weight. Global routing identifies the scale preference of the current image, whereas local routing refines that preference spatially, allowing small gaps, ablation traces, and background regions to follow different fusion policies.

The weighted feature is first mixed and injected into the target-level feature through a gated residual. The target feature remains an uninterrupted information path so that cross-scale responses do not destabilize the original representation early in training. A 3×3 depthwise convolution and a 1×1 pointwise convolution then refine the fused feature, and a second learnable gate controls the refinement magnitude. Both gates use sigmoid activation, and their values determine how much cross-scale information and local refinement are added to the target feature.

Following bidirectional aggregation and ORI-ELAN integration, BRG-FPN outputs O_3, O_4, and O_5. By preserving the target level while selecting cross-scale evidence according to image content and position, the pyramid provides stable representations of local defects at different resolutions and semantically consistent inputs to text attention and the TG-D-FINE decoder.

4. Experiments and Analysis

4.1. Dataset

The experimental images were collected from the Chinese Power Line Insulator Dataset (CPLID) [26], the STN Power Line Assets Dataset (STN PLAD) [27], and field inspections. All samples are visible-light images of transmission-line insulators. They vary substantially in object scale, insulator pose, illumination, and background composition and include distant small targets, partial occlusion, vegetation, towers, conductors, and other clutter typical of transmission-line inspection. After harmonization, the dataset contains 1600 images in total: 773 from CPLID, 519 from STN PLAD, and 308 from field inspections. The dataset was divided at the image level into 1120 training images, 320 validation images, and 160 test images using a 7:2:1 ratio.

Differences in class names and annotation extents across the source datasets were resolved under a common class taxonomy and bounding-box protocol. Annotations follow the Common Objects in Context (COCO) detection format and include broken, flashover, and insulator. Broken covers shed gaps, missing material, and localized structural damage; flashover denotes visible discharge traces, ablation, or carbonized regions; and insulator denotes the complete object. Broken and flashover boxes tightly enclose the visible anomaly, whereas insulator boxes cover the entire device. Because a localized defect is spatially contained within the insulator, object-level and defect-level boxes may coexist for the same physical target. The final dataset contains 4429 instances, comprising 1165 broken, 1377 flashover, and 1887 insulator instances, accounting for 26.3%, 31.1%, and 42.6% of the total, respectively.

Representative samples of the three categories are shown in Figure 6.



(a)Intact insulator (b)Broken defect (c)Flashover defect

Figure 6. Representative samples of the three detection categories.

4.2. Experimental Setup

Model training and workstation-based evaluations were conducted in the hardware and software environment listed in Table 1. Input images were resized to 640 × 640 pixels, and training lasted 300 epochs with a batch size of 4. The Adam optimizer with decoupled weight decay (AdamW) [28] was used with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. The learning rate followed a FlatCosine schedule. Unless otherwise specified, all models used the same dataset split, training settings, and evaluation procedure.

Table 1. Experimental environment.

Configuration	Setting
Operating system	Windows 11, 64-bit
CPU	AMD Ryzen 9 7945HX
GPU	NVIDIA GeForce RTX 5080
Deep learning framework	PyTorch 2.12.0
CUDA version	CUDA 13.2
Python version	Python 3.11

Online data augmentation included photometric distortion, random zoom-out, cropping based on intersection over union (IoU), and horizontal flipping to reduce the risk of overfitting. Only horizontal flipping was retained during the final 12 epochs. Validation and test images were resized to 640 × 640 pixels without random augmentation.

4.3. Evaluation Metrics

Detection performance was evaluated using precision, recall, mean average precision at an IoU threshold of 0.50 (mAP@50), and mAP averaged over IoU thresholds from 0.50 to 0.95 in increments of 0.05 (mAP@50–95). Precision measures the proportion of predicted targets that are correct, whereas recall measures the proportion of ground-truth targets that are detected. Average precision (AP) is the area under the precision–recall curve, and mean average precision (mAP) averages AP over all classes. The definitions are given in Equations (8)–(12). Model complexity and computational cost were reported using parameter count and floating-point operation counts in billions (GFLOPs), respectively. Inference efficiency was evaluated using frames per second (FPS) and model inference latency.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$AP = \int_0^1 P(R)dR \tag{10}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{11}$$

$$mAP_{50-95} = \frac{1}{10}\sum_{k=0}^{9} mAP_{0.50+0.05k} \tag{12}$$

where TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, $P(R)$ denotes precision as a function of recall, N is the number of classes, and i indexes the class. The index k identifies one of the ten IoU thresholds.

4.4. Ablation Study

Ablation experiments were conducted to assess the contribution of text guidance and each visual component. Starting from DEIM-D-FINE-S, the offline text-guided branch, DPStem, ORI-ELAN, and BRG-FPN were introduced individually and in combination under identical data splits, training schedules, and evaluation settings. The results are summarized in Table 2. Mean denotes mAP@50, while Ins., Br., and Fl. denote insulator, broken, and flashover, respectively.

Table 2. Ablation results for the Detail-Preserving, Routing-Aware, and Bidirectional-Fusion DEIM (DRB-DEIM).

Model	P/%	R/%	AP@50/%				mAP@50–95/%	Params/M	FPS
			Mean	Ins.	Br.	Fl.			
Base (DEIM-D-FINE-S)	89.1	79.2	82.8	96.1	76.6	75.8	50.3	10.18	128.8
+ TG (TG-DEIM-D-FINE-S)	88.9	80.8	84.1	96.5	79.7	76.1	53.6	11.70	112.1
TG + DPStem	90.5	81.9	85.3	97.1	82.5	76.3	54.1	11.67	113.4
TG + ORI-ELAN	91.3	81.2	85.6	97.2	82.9	76.7	54.6	11.61	115.7
TG + BRG-FPN	89.8	83.4	86.9	98.5	85.2	77.0	55.2	11.53	118.4
TG + DPStem + ORI-ELAN	90.1	83.1	87.1	98.2	86.0	77.1	57.2	11.61	118.3
TG + DPStem + BRG-FPN	89.6	83.2	86.8	98.5	85.3	76.6	56.9	11.52	117.7
TG + ORI-ELAN + BRG-FPN	90.3	82.9	87.4	98.3	86.5	77.4	57.4	11.57	117.8
DRB-DEIM	89.9	83.5	88.1	99.1	87.6	77.6	59.2	11.57	117.4

Adding offline text guidance increases recall from 79.2% to 80.8%, mAP@50 from 82.8% to 84.1%, and mAP@50–95 from 50.3% to 53.6%. Precision changes slightly from 89.1% to 88.9%. The class-wise results show a larger gain for broken, whose AP@50 rises from 76.6% to 79.7%, compared with increases of 0.4 and 0.3 percentage points for insulator and flashover, respectively. These gains support the effectiveness of the integrated text-guided design, which combines pretrained class references with trainable adaptation and similarity-based attention.

Each visual module improves detection performance over the Text-Guided DEIM-D-FINE-S (TG-DEIM-D-FINE-S) baseline. DPStem reaches 85.3% mAP@50 with nearly unchanged parameter count, supporting the benefit of preserving shallow defect evidence. ORI-ELAN produces the highest single-module precision of 91.3% and raises mAP@50–95 to 54.6%, indicating that orientation routing and local-contrast modeling improve morphological discrimination. BRG-FPN gives the strongest single-module overall result, achieving 83.4% recall, 86.9% mAP@50, and 55.2% mAP@50–95 while increasing throughput to 118.4 FPS. It also gives the highest single-module AP@50 for broken and flashover, at 85.2% and 77.0%, respectively.

The paired configurations further show that the modules are complementary. DPStem with ORI-ELAN achieves 87.1% mAP@50 and 57.2% mAP@50–95. The combination of ORI-ELAN and BRG-FPN provides the highest mAP@50 and mAP@50–95 among the two-module configurations, reaching 87.4% and 57.4%, respectively. These results indicate that orientation-aware transformation and adaptive cross-scale fusion reinforce local defect representations at successive stages. However, the class-wise gains are not uniform: adding DPStem to TG + BRG-FPN reduces flashover AP@50 from 77.0% to 76.6%, despite increasing mAP@50–95 from 55.2% to 56.9%.

The complete DRB-DEIM achieves the highest mAP@50 and mAP@50–95, at 88.1% and 59.2%, respectively. Relative to TG-DEIM-D-FINE-S, these metrics improve by 4.0 and 5.6 percentage points, while precision and recall reach 89.9% and 83.5%. Compared with the original visual baseline, the gains are 5.3 and 8.9 percentage points. Class-wise AP@50 reaches 99.1% for insulator, 87.6% for broken, and 77.6% for flashover, corresponding to improvements of 3.0, 11.0, and 1.8 percentage points, respectively, over the original visual baseline. The complete design therefore improves both defect retrieval and boundary localization, with the largest class-wise gain observed for broken and a more modest improvement for flashover.

DRB-DEIM contains 11.57 M parameters, approximately 0.13 M fewer than TG-DEIM-D-FINE-S, while increasing throughput from 112.1 to 117.4 FPS. Relative to the original visual baseline, it adds approximately 1.39 M parameters.

Table 3 summarizes the computational cost and inference latency at an input resolution of 640 × 640 pixels. Groups 1–9 follow the configuration order in Table 2. For these ablation configurations, latency denotes the average model inference time per image, excluding preprocessing and postprocessing. Relative to TG-DEIM-D-FINE-S, ORI-ELAN and BRG-FPN reduce the computational cost from 26.2 GFLOPs to 25.6 and 25.78 GFLOPs, respectively, while inference latency decreases from 8.02 ms to 7.74 and 7.54 ms. These reductions are consistent with the partial-channel depthwise design of ORI-ELAN and the fixed-width fusion of BRG-FPN. The complete DRB-DEIM requires 25.47 GFLOPs and achieves an inference latency of 7.61 ms. The differing GFLOPs and latency rankings across configurations show that arithmetic cost alone does not determine inference speed.

Table 3. Computational cost and inference latency of the ablation configurations.

Metric	1	2	3	4	5	6	7	8	9
GFLOPs	24.83	26.2	25.84	25.6	25.78	25.24	25.42	25.83	25.47
Latency/ms	6.86	8.02	7.91	7.74	7.54	7.54	7.59	7.58	7.61

4.5. Comparison with Representative Detectors

Comparison experiments were conducted to evaluate DRB-DEIM against representative visual and vision–language detectors. The conventional single-stage group includes YOLOv8-S [29], YOLO11-S [30], YOLO12-S [31], and Gold-YOLO-S [32]. YOLO-World-S [22] and YOLOE-11-S [23] were included as other vision–language detectors to extend the comparison to models incorporating text semantics. RT-DETR [12] and RT-DETRv2 [13] were selected as end-to-end Transformer detectors, while DEIM-D-FINE-S and DEIM-D-FINE-M [14,15] were used as the direct baselines. All models were trained and evaluated with the same dataset split and experimental settings. TensorRT acceleration was not used for the comparisons in Table 4.

The two vision–language detectors achieve higher mAP@50 and mAP@50–95 than the conventional YOLO models on this dataset. YOLO-World-S reaches 83.3% mAP@50 and 51.7% mAP@50–95 with 12.76 M parameters and a throughput of 115.3 FPS. YOLOE-11-S obtains slightly lower values of 82.9% and 51.3%, respectively, but provides higher recall and throughput with fewer parameters.

RT-DETR achieves 84.3% mAP@50 and 51.1% mAP@50–95 with 19.87 M parameters. RT-DETRv2 raises these values to 84.9% and 53.4%, respectively, while retaining a similar parameter count. Its throughput also increases from 79.3 to 90.1 FPS. Both Transformer detectors outperform the conventional YOLO and vision–language baselines in mAP@50, but their parameter counts remain close to 20 M and their inference speeds are lower than those of most lightweight models.

Table 4. Comparison of DRB-DEIM with representative object detectors.

Model	Precision/%	Recall/%	mAP@50/%	mAP@50–95/%	Params/M	FPS
YOLOv8-S	81.6	72.5	78.5	46.8	11.13	85.9
YOLO11-S	84.5	75.2	81.1	50.3	9.41	122.5
YOLO12-S	82.4	75.0	80.6	50.4	9.08	104.8
Gold-YOLO-S	83.3	76.1	80.9	49.6	21.50	108.1
YOLO-World-S	85.7	77.1	83.3	51.7	12.76	115.3
YOLOE-11-S	82.1	80.1	82.9	51.3	9.43	121.4
RT-DETR	86.0	77.0	84.3	51.1	19.87	79.3
RT-DETRv2	83.2	79.4	84.9	53.4	19.88	90.1
DEIM-D-FINE-S	89.1	79.2	82.8	50.3	10.18	128.8
DEIM-D-FINE-M	88.7	81.8	85.0	52.7	19.19	98.3
DRB-DEIM	89.9	83.5	88.1	59.2	11.57	117.4

DEIM-D-FINE-S achieves the highest throughput among the compared methods at 128.8 FPS, but its recall and mAP@50–95 are 79.2% and 50.3%. Scaling the detector to DEIM-D-FINE-M raises mAP@50 to 85.0% and mAP@50–95 to 52.7%; however, the parameter count increases to 19.19 M and throughput falls to 98.3 FPS. Increasing capacity alone therefore yields limited accuracy improvement at a clear efficiency cost.

DRB-DEIM achieves 89.9% precision, 83.5% recall, 88.1% mAP@50, and 59.2% mAP@50–95, giving the highest values for all four accuracy measures. Compared with DEIM-D-FINE-S, mAP@50 and mAP@50–95 increase by 5.3 and 8.9 percentage points, respectively, while the parameter count increases by approximately 1.39 M. The model retains an inference speed of 117.4 FPS. Relative to YOLO-World-S and YOLOE-11-S, DRB-DEIM improves mAP@50 by 4.8 and 5.2 percentage points and mAP@50–95 by 7.5 and 7.9 percentage points, respectively. Compared with DEIM-D-FINE-M, it uses 39.7% fewer parameters and improves mAP@50 and mAP@50–95 by 3.1 and 6.5 percentage points. These results support the effectiveness of the integrated text guidance and visual redesign for improving insulator defect detection while maintaining high inference efficiency.

4.6. Evaluation Under Synthetic Image Degradations

The original dataset includes images with environmental interference. Synthetic degradations were additionally applied to provide controlled comparisons on the same test scenes and annotations. Six degraded test sets were constructed from the original test images to examine model robustness. The degradation types include ISO noise, occlusion, rain, snow, fog, and glare. ISO noise reproduces the luminance and chromatic fluctuations that may occur when a UAV camera operates at high sensitivity or under low illumination. Only the image appearance was changed, while the original class labels and bounding boxes were retained. The trained models were evaluated directly on these sets without retraining or fine-tuning. Examples of the degraded images are shown in Figure 7, and the results are reported in Table 5. The degradation settings are provided in Table S1.

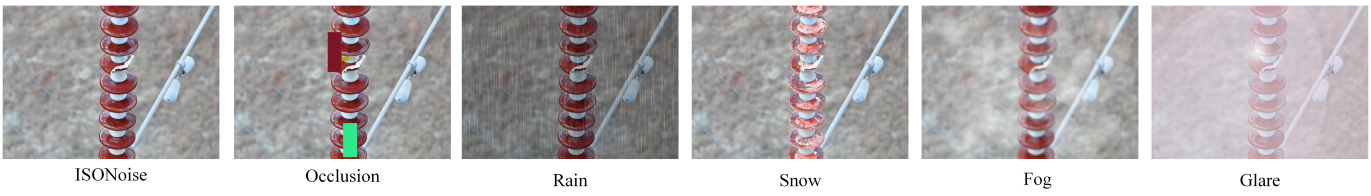


Figure 7. Examples of the six image degradations. Colored rectangles in the occlusion example represent synthetic occluders and do not indicate object classes.

Table 5. Detection performance under synthetic image degradations.

Model	mAP@50/%					
	ISO Noise	Occlusion	Rain	Snow	Fog	Glare
YOLOv8-S	73.9	58.6	66.5	39.2	48.9	54.0
YOLO12-S	75.1	61.9	63.7	49.5	55.2	32.3
YOLO11-S	74.3	62.1	62.2	51.6	57.5	35.1
Gold-YOLO-S	74.7	60.3	64.1	47.2	54.4	37.5
YOLO-World-S	75.5	62.7	68.2	51.9	56.5	40.9
YOLOE-11-S	75.1	62.6	65.0	52.2	54.1	48.1
RT-DETR	76.2	64.0	75.6	50.1	26.6	53.2
RT-DETRv2	77.5	62.8	77.3	51.4	23.8	78.7
DEIM-D-FINE-S	77.8	62.3	71.2	51.2	48.8	73.9
DRB-DEIM	84.6	71.4	77.2	59.4	54.6	82.7

The six degradation types affect the compared models differently. Occlusion, snow, and fog cause substantial accuracy losses for most methods. The conventional YOLO models retain relatively high mAP@50 under ISO noise but show larger variations under snow and glare. RT-DETRv2 performs well under rain and glare, reaching 77.3% and 78.7% mAP@50, respectively. Its accuracy decreases to 23.8% under fog, indicating sensitivity to reduced contrast and weakened surface texture.

The two vision–language detectors also exhibit different robustness profiles. YOLO-World-S achieves higher mAP@50 than YOLOE-11-S under ISO noise, occlusion, rain, and fog. Under fog, YOLO-World-S reaches 56.5%, approaching the highest result of 57.5% obtained by YOLO11-S. YOLOE-11-S performs better under snow and glare, reaching 52.2% and 48.1%, respectively, compared with 51.9% and 40.9% for YOLO-World-S.

DRB-DEIM achieves the highest mAP@50 under ISO noise, occlusion, snow, and glare, with values of 84.6%, 71.4%, 59.4%, and 82.7%, respectively. These results exceed those of the second-best model by 6.8, 7.4, 7.2, and 4.0 percentage points under the corresponding conditions. Under rain, DRB-DEIM reaches 77.2% mAP@50, only 0.1 percentage point below RT-DETRv2. Under fog, it obtains 54.6%, remaining below YOLO11-S and YOLO-World-S but exceeding DEIM-D-FINE-S by 5.8 percentage points. The proposed model therefore provides the strongest results under four degradation types, while its advantage is less pronounced under rain and fog.

Averaged across the six degraded sets, DRB-DEIM obtains the highest mAP@50 of 71.7%. DEIM-D-FINE-S, YOLO-World-S, and YOLOE-11-S achieve 64.2%, 59.3%, and 59.5%, respectively, corresponding to average gains of 7.5, 12.4, and 12.1 percentage points for DRB-DEIM. The improvement over DEIM-D-FINE-S is observed under all six conditions, supporting the effectiveness of the integrated design for maintaining insulator defect detection performance under different image degradations.

4.7. Deployment Efficiency Analysis

Deployment experiments were conducted on an NVIDIA Jetson Nano B01 with 4 GB of memory to evaluate the feasibility of DRB-DEIM for local insulator defect detection. Three inference configurations were compared: PyTorch 32-bit floating-point precision (FP32), TensorRT FP32, and TensorRT 16-bit floating-point precision (FP16). The software environment comprised Ubuntu 18.04.6 LTS, JetPack 4.6.1, CUDA 10.2, cuDNN 8.2.1.32, PyTorch 1.10, TensorRT 8.2.1, and Python 3.6.9. The input resolution was 640 × 640, and the inference batch size was set to 1. Board-level power consumption was measured using the power rail at 100 ms intervals and averaging the readings over the timed inference period. Latency was measured for model inference only, excluding preprocessing and post-

processing. Detection accuracy, model inference speed, board-level power consumption, and model latency are summarized in Table 6.

Table 6. Detection accuracy, model inference speed, board-level power consumption, and model latency of DRB-DEIM on the NVIDIA Jetson Nano B01 4 GB.

Settings	mAP@50/%	mAP@50–95/%	FPS	Power/W	Latency/ms
PyTorch (FP32)	88.0	59.0	11.06	14.68	90.41
TensorRT (FP32)	87.8	58.7	16.27	10.22	61.47
TensorRT (FP16)	87.3	58.5	20.83	11.56	48.01

The PyTorch FP32 configuration achieves 88.0% mAP@50 and 59.0% mAP@50–95. After conversion to TensorRT FP32, these values change to 87.8% and 58.7%, respectively, corresponding to decreases of 0.2 and 0.3 percentage points. TensorRT FP16 obtains 87.3% mAP@50 and 58.5% mAP@50–95, representing decreases of 0.7 and 0.5 percentage points relative to PyTorch FP32.

TensorRT deployment improves model inference speed on the edge platform. Throughput increases from 11.06 FPS with PyTorch FP32 to 16.27 FPS with TensorRT FP32, while model latency decreases from 90.41 to 61.47 ms. TensorRT FP16 further increases throughput to 20.83 FPS and reduces model latency to 48.01 ms. Relative to PyTorch FP32, the FP16 configuration provides an 88.3% increase in throughput and a 46.9% reduction in latency.

The measured board-level power decreases from 14.68 W with PyTorch FP32 to 10.22 W with TensorRT FP32, a reduction of 30.4%. TensorRT FP16 consumes 11.56 W, which remains 21.3% below the PyTorch configuration but is 13.1% higher than TensorRT FP32. Thus, TensorRT FP32 gives the lowest measured board-level power and higher detection accuracy than TensorRT FP16, whereas FP16 provides the highest throughput and lowest model latency. These results demonstrate the feasibility of running DRB-DEIM on the Jetson Nano for local insulator defect inference.

5. Visualization Analysis

5.1. ORI-ELAN Routing Coefficients

The routing coefficients of ORI-ELAN were visualized to examine how the branch weights change with the input. The same image was rotated by 0°, 45°, and 90° on a common canvas, with the image scale kept consistent. All coefficients were extracted from the same ORI block at the P3 level during inference. Figure 8 presents the three inputs and their corresponding coefficients. Colors distinguish the four branches, while marker shapes identify the three rotation angles.

The identity coefficient increases from 0.9738 at 0° to 1.0528 at 45° and 1.0688 at 90°. The local coefficient changes more moderately, increasing from 1.0049 to 1.0120. Over the same rotation range, the horizontal and vertical coefficients decrease from 1.0032 and 0.9807 to 0.9932 and 0.9540, respectively. These changes show that the router adjusts the balance between the identity pathway and the spatial transformations as the input rotates.

The variation reflects the router's response to the spatial arrangement of the insulator and its surrounding scene. Most coefficients remain close to 1, indicating moderate adjustment of branch amplitudes. All four branches remain active and contribute spatially resolved features. The visualization therefore illustrates the intended behavior of ORI-ELAN: the router adjusts the overall balance of local and directional responses according to the image context, while the convolutional branches continue to represent local structures.

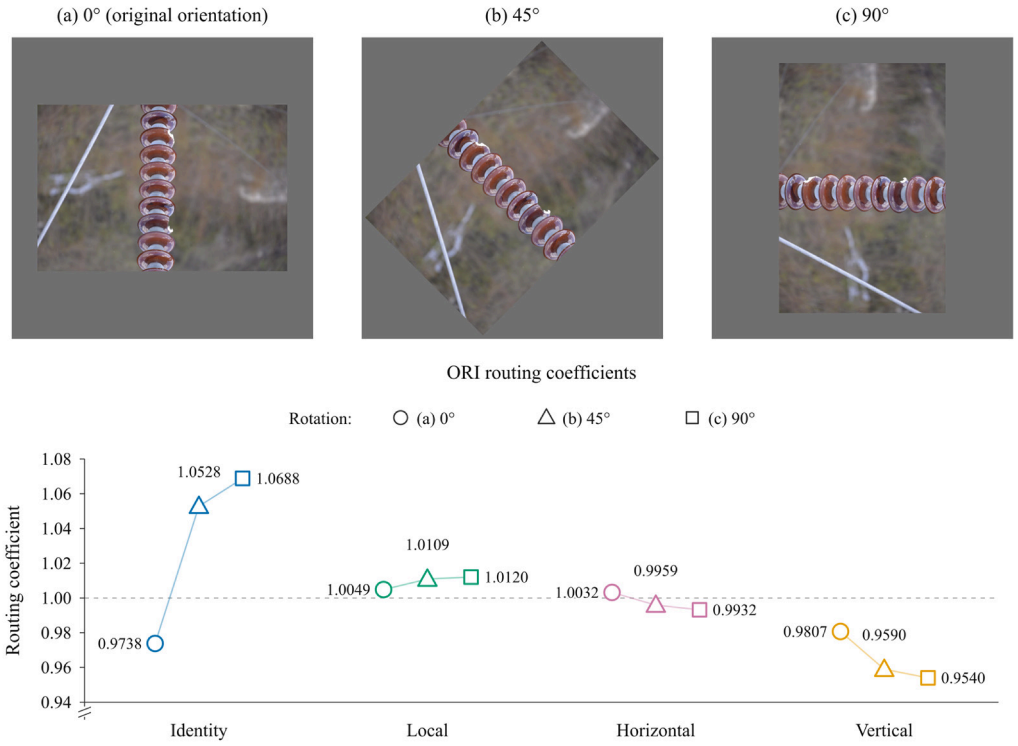


Figure 8. ORI-ELAN routing coefficients for the same image at different rotation angles.

5.2. Inference Analysis

Detection examples from a challenging scene were compared to examine differences among the models. Broken sheds usually have clear gaps or discontinuous edges and were detected by most methods, resulting in only small visual differences. Flashover marks are more difficult to distinguish because they are often small, irregular, and affected by illumination and surface reflection. Figure 9 therefore presents a scene containing several flashover regions with different sizes and levels of visibility.



Figure 9. Detection results of different models in a scene containing multiple flashover defects.

Most models detect the prominent flashover regions along the insulator, while their performance differs on smaller and less visible defects. YOLOE-11-S and YOLO-World-S identify several flashover regions distributed across the insulator, including the small mark

on the left side below the uppermost shed. However, both miss the small flashover in the upper-right area near the metal fitting. This region is detected only by RT-DETRv2, DEIM-D-FINE-S, and DRB-DEIM in the illustrated comparison. Its limited spatial extent and weak contrast make it difficult to distinguish from the surrounding surface and background.

The flashover at the bottom shows only a slight color change and has no clear structural boundary, making it difficult to distinguish from the normal insulator surface. Only DRB-DEIM detects this region. In this scene, DRB-DEIM detects the prominent middle defects as well as the small upper-right and faint lower flashover regions.

The improved detections are consistent with the intended roles of the proposed modules. DPStem retains weak textures during early downsampling, ORI-ELAN captures irregular ablation boundaries under different target orientations, and BRG-FPN combines shallow localization information with deeper semantic features. Text guidance also helps distinguish faint flashover traces from the normal insulator surface. These observations are consistent with the higher mAP@50–95 of DRB-DEIM.

Figure 10 presents two representative failure cases of DRB-DEIM. In Figure 10a, the confidence threshold was set to 0.75. The more visible flashover regions are detected, whereas smaller traces are missed at this threshold. Their limited spatial extent and less distinct appearance may make them difficult to detect with high confidence.

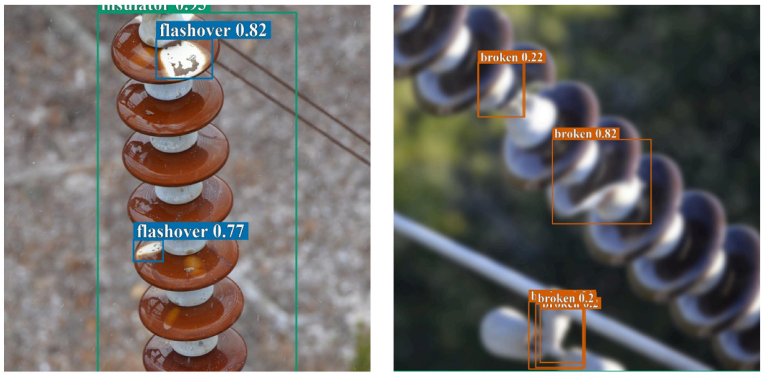


Figure 10. Representative failure cases of DRB-DEIM: (**a**) missed detection of small flashover traces; (**b**) false identification of a conductor-associated component as broken under synthetic fog.

In Figure 10b, a conductor fitting near the lower image boundary is incorrectly classified as broken under synthetic fog. Reduced contrast and blurred contours obscure its structural details, making it harder to distinguish from damaged insulator edges. The visual similarity between the fitting and local structural damage may explain this false positive. These cases highlight the remaining challenges of detecting small flashover traces and separating defects from background fittings under reduced visibility.

Although small flashover traces and background fittings remain challenging, DRB-DEIM provides more complete defect detection in the illustrated comparison. This observation is consistent with its higher overall recall and mAP@50–95 on the test set.

5.3. Heatmap Analysis

Gradient-weighted Class Activation Mapping (Grad-CAM) [33] was used to compare the internal responses of DEIM-D-FINE-S, TG-DEIM-D-FINE-S, and DRB-DEIM. Figure 11 presents the broken example in the first row and the flashover example in the second row. Red and yellow indicate strong activation, whereas blue indicates weak activation.

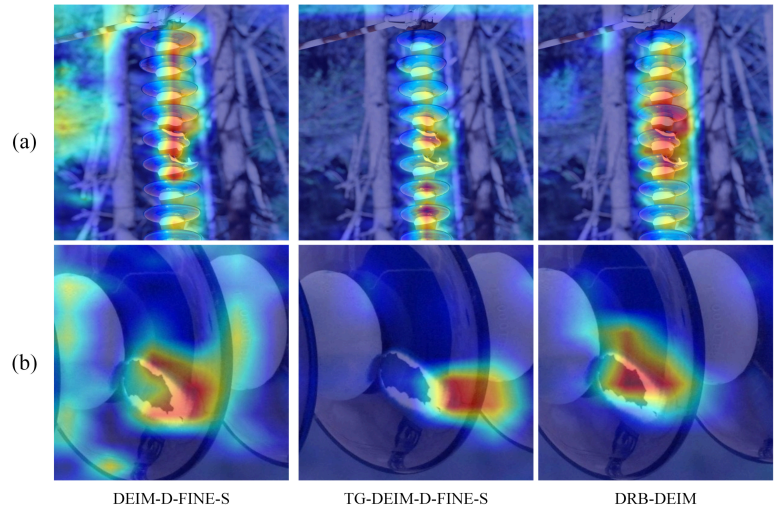


Figure 11. Grad-CAM responses of different models to (**a**) broken and (**b**) flashover defects. Red and yellow indicate stronger activation, whereas blue indicates weaker activation.

For the broken example, DEIM-D-FINE-S produces strong activation on the insulator, but substantial responses also appear in the surrounding background. The activation does not clearly distinguish the damaged shed from nearby intact structures. TG-DEIM-D-FINE-S markedly suppresses these background responses and directs more attention toward the insulator region, although its activation around the break remains relatively dispersed. DRB-DEIM concentrates its peak activation around the damaged sheds and their discontinuous boundaries, with weaker responses in unrelated background regions.

A similar difference is observed for flashover. DEIM-D-FINE-S responds strongly to the visible trace, but its activation spreads across the adjacent insulator surface and several background regions. Text guidance substantially reduces this interference in TG-DEIM-D-FINE-S; however, the peak response is displaced from the actual flashover region toward the neighboring intact surface. The peak response of DRB-DEIM is more closely aligned with the visible flashover trace. Its activation also follows the irregular defect boundary more closely.

These examples show that activation magnitude and spatial alignment should be considered together. TG-DEIM-D-FINE-S suppresses much of the background activation, while DRB-DEIM shows closer alignment with the visible defects. This pattern is consistent with the detection examples and the higher overall recall and mAP@50–95 of DRB-DEIM.

6. Conclusions

This study developed DRB-DEIM for detecting localized defects in UAV images of transmission-line insulators. The model combines offline TIPSv2 class prototypes with three visual components. DPStem preserves local information during early downsampling, ORI-ELAN adjusts directional responses according to image context, and BRG-FPN aggregates localization and semantic information across scales. The cached prototypes provide shared class references for feature calibration, query selection, and classification without running the text encoder during inference.

DRB-DEIM achieves 89.9% precision, 83.5% recall, 88.1% mAP@50, and 59.2% mAP@50–95 with 11.57 M parameters and an inference speed of 117.4 FPS. Compared with DEIM-D-FINE-S, mAP@50 and mAP@50–95 increase by 5.3 and 8.9 percentage points, respectively. The larger gain in mAP@50–95 is consistent with improved detection under stricter overlap requirements. DRB-DEIM also achieves higher detection accuracy than DEIM-D-FINE-M while using 39.7% fewer parameters, showing that the proposed changes are more effective on this dataset than increasing the baseline model scale.

The ablation results support the contributions of text guidance and the three visual components. Text guidance improves recall and mAP@50–95, while the visual modules provide further gains through complementary changes to feature extraction and fusion. The class-wise improvements are more pronounced for broken defects than for flashover traces. Detection examples and Grad-CAM maps provide supporting qualitative evidence, showing more complete detection of weak defects and closer alignment between activation and visible abnormal regions. Small flashover traces and background fittings under reduced visibility remain challenging.

Across six synthetically degraded test sets, DRB-DEIM obtains an average mAP@50 of 71.7%, exceeding DEIM-D-FINE-S by 7.5 percentage points. It achieves the highest accuracy under ISO noise, occlusion, snow, and glare, although other detectors perform better under rain and fog. These results support improved detection stability under controlled image degradations. The Jetson Nano experiments further confirm the feasibility of deploying DRB-DEIM on an embedded platform for local processing of inspection images.

Future work will investigate structured pruning and knowledge distillation to reduce the computational and storage requirements of the 11.57 M-parameter model. Evaluation will also be extended from still images and model-only inference to complete processing pipelines using continuous UAV inspection videos. Additional data from adverse weather, low illumination, and different cameras will support further investigation of adaptive prompt learning and cross-domain feature alignment.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app162010297/s1, Table S1: Parameter settings for the six synthetic image degradations.

Author Contributions: Conceptualization, H.W.; methodology, H.W.; software, H.W.; validation, H.W. and R.J.; formal analysis, H.L. and H.W.; investigation, H.W. and R.J.; resources, H.L. and H.W.; data curation, H.L. and H.W.; writing—original draft preparation, H.W.; writing—review and editing, H.L., H.W. and R.J.; visualization, H.W.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52277108.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The publicly available datasets used in this study can be accessed from the Chinese Power Line Insulator Dataset (CPLID) at https://github.com/InsulatorData/InsulatorDataSet (accessed on 27 September 2025) and the STN Power Line Assets Dataset (STN PLAD) at https://github.com/andreluizbvs/PLAD (accessed on 29 September 2025). The field inspection images and the harmonized COCO-format annotations generated in this study are available from the corresponding author upon reasonable request. The field data are not publicly available because of data-ownership restrictions.

Acknowledgments: The authors would like to thank the personnel involved in the field inspection and image collection for their assistance with data acquisition.

Conflicts of Interest: The authors declare no conflicts of interest. The funder had no role in the design of the study; in the collection, analysis, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DRB-DEIM	Detail-Preserving, Routing-Aware, and Bidirectional-Fusion DEIM
TG	Text-Guided
DEIM	DETR with Improved Matching
D-FINE	Fine-Grained Distribution Refinement
DPStem	Detail-Preserving Stem
ORI-ELAN	Orientation-Routed Inception Efficient Layer Aggregation Network
BRG-FPN	Bidirectional Routing-Guided Feature Pyramid Network
BRGFusion	Bidirectional Routing-Guided Fusion
TIPSv2	Text–Image Pretraining with Spatial Awareness, Version 2
ORI	Orientation-Routed Inception
DETR	Detection Transformer
HGNetV2	High-Performance GPU Network V2
SCDown	Spatial–Channel Decoupled Downsampling
RMSNorm	Root Mean Square Layer Normalization
FiLM	Feature-wise Linear Modulation
FP32	32-Bit Floating-Point Precision
FP16	16-Bit Floating-Point Precision
RT-DETR	Real-Time Detection Transformer
CLIP	Contrastive Language–Image Pretraining
YOLO	You Only Look Once
FPN	Feature Pyramid Network
ELAN	Efficient Layer Aggregation Network
Grad-CAM	Gradient-Weighted Class Activation Mapping
IoU	Intersection over Union
AP	Average Precision
mAP	Mean Average Precision
FPS	Frames Per Second
AdamW	Adaptive Moment Estimation with Decoupled Weight Decay
COCO	Common Objects in Context
CPLID	Chinese Power Line Insulator Dataset
STN PLAD	STN Power Line Assets Dataset
UAV	Unmanned Aerial Vehicle
RGB	Red, Green, and Blue

References

1. 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* **2023**, *23*, 5249. [CrossRef] [PubMed]
2. Lu, Z.; Li, Y.; Shuang, F. MGFNet: A Progressive Multi-Granularity Learning Strategy-Based Insulator Defect Recognition Algorithm for UAV Images. *Drones* **2023**, *7*, 333. [CrossRef]
3. Wang, X.; Yang, T.; Zou, Y. Enhancing Grid Reliability through Advanced Insulator Defect Identification. *PLoS ONE* **2024**, *19*, e0307684. [CrossRef] [PubMed]
4. Liu, S.; Zhang, W.; Yuan, S.; Bao, H.; Mao, W.; Xi, S. A Lightweight Model for Insulator Defect Detection Based on Vision–Language Modeling and Prior Knowledge in Power Systems. *Processes* **2025**, *13*, 3714. [CrossRef]
5. Cao, L.; Zhao, J.; Qiao, L.; Zhai, X.; Zhang, L. BEAD-Net: Bidirectional Fusion and Hourglass Expanded Asymmetric Detection for Insulator Defect Detection. *Electronics* **2026**, *15*, 3580. [CrossRef]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
7. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125. [CrossRef]

8. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. [CrossRef]
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229. [CrossRef]
10. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
11. 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 Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
12. 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, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974. [CrossRef]
13. Lv, W.; Zhao, Y.; Chang, Q.; Huang, K.; Wang, G.; Liu, Y. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer. *arXiv* **2024**, arXiv:2407.17140. [CrossRef]
14. Peng, Y.; Li, H.; Wu, P.; Zhang, Y.; Sun, X.; Wu, F. D-FINE: Redefine Regression Task of DETRs as Fine-Grained Distribution Refinement. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
15. Huang, S.; Lu, Z.; Cun, X.; Yu, Y.; Zhou, X.; Shen, X. DEIM: DETR with Improved Matching for Fast Convergence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 11–15 June 2025; pp. 15162–15171. [CrossRef]
16. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 18–24 July 2021*; Meila, M., Zhang, T., Eds.; Proceedings of Machine Learning Research; PMLR: London, UK, 2021; Volume 139, pp. 8748–8763.
17. Cao, B.; Chen, K.; Maninis, K.-K.; Chen, K.; Karpur, A.; Xia, Y.; Dua, S.; Dabral, T.; Han, G.; Han, B.; et al. TIPSv2: Advancing Vision–Language Pretraining with Enhanced Patch–Text Alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Denver, CO, USA, 3–7 June 2026; pp. 29325–29335.
18. Gu, X.; Lin, T.-Y.; Kuo, W.; Cui, Y. Open-Vocabulary Object Detection via Vision and Language Knowledge Distillation. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022.
19. Zhong, Y.; Yang, J.; Zhang, P.; Li, C.; Codella, N.; Li, L.H.; Zhou, L.; Dai, X.; Yuan, L.; Li, Y.; et al. RegionCLIP: Region-Based Language–Image Pretraining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 16793–16803. [CrossRef]
20. Li, L.H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. Grounded Language–Image Pre-Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 10965–10975. [CrossRef]
21. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *Computer Vision—ECCV 2024*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15105, pp. 38–55. [CrossRef]
22. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. YOLO-World: Real-Time Open-Vocabulary Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16901–16911. [CrossRef]
23. Wang, A.; Liu, L.; Chen, H.; Lin, Z.; Han, J.; Ding, G. YOLOE: Real-Time Seeing Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Honolulu, HI, USA, 19–23 October 2025; pp. 24591–24602. [CrossRef]
24. Zhang, K.; Zheng, Z.; Wang, J.; Yang, J.; Xiao, Y. Multimodal Knowledge-Guided Method for Power Transmission Line Fault Detection Using a Vision–Language Model. *Int. J. Electr. Power Energy Syst.* **2026**, *176*, 111696. [CrossRef]
25. Chen, Z.; Guo, F.; Chen, Z.; Zuo, L.; Liu, Y. PCBClip: Vision–Language Defect Detection Model for Low-Sample Inspection Systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 13406–13423. [CrossRef] [PubMed]
26. Tao, X.; Zhang, D.; Wang, Z.; Liu, X.; Zhang, H.; Xu, D. Detection of Power Line Insulator Defects Using Aerial Images Analyzed with Convolutional Neural Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 1486–1498. [CrossRef]
27. Vieira-e-Silva, A.L.B.; Felix, H.C.; Chaves, T.M.; Simões, F.P.M.; Teichrieb, V.; dos Santos, M.M.; Santiago, H.C.; Sgotti, V.A.C.; Lott Neto, H.B.D.T. STN PLAD: A Dataset for Multi-Size Power Line Assets Detection in High-Resolution UAV Images. In Proceedings of the 2021 34th SIBGRAPI Conference on Graphics, Patterns and Images, Gramado, Brazil, 18–22 October 2021; pp. 215–222. [CrossRef]
28. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

29. Ultralytics. Ultralytics YOLOv8. Available online: https://docs.ultralytics.com/models/yolov8/ (accessed on 20 September 2026).
30. Ultralytics. Ultralytics YOLO11. Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 20 September 2026).
31. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *Adv. Neural Inf. Process. Syst.* **2025**, *38*, 78433–78457. [CrossRef]
32. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient Object Detector via Gather-and-Distribute Mechanism. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 51094–51112. [CrossRef]
33. 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 Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.