Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 10(2): 205–239.
DOI: 10.3934/electreng.2026009
Received: 20 September 2025
Revised: 08 January 2026
Accepted: 03 February 2026
Published: 04 March 2026

https://www.aimspress.com/journal/ElectrEng

Research article

A lightweight frequency-aware enrichment module (FAEM) architecture for tiny insulator defect detection

Christopher D. Naya[1,*], Elvis Twumasi[1], Eliel Keelson[2] and Abdul-Majid Issah Malori[1]

[1] Electrical/Electronic Engineering Department, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
[2] Computer Engineering Department, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

* **Correspondence:** Email: chrisdnaya@gmail.com; Tel.: +233 554390510.

Abstract: Real-time detection of tiny insulator defects is critical for power grid reliability. However, it remains a major challenge for Unmanned Aerial Vehicle (UAV) based inspection systems due to the trade-off between model accuracy and computational efficiency. To address this challenge, we proposed a novel lightweight object detector that incorporates a Frequency-Aware Enrichment Module (FAEM) into a You Only Look Once (YOLO) architecture for tiny insulator defect detection. FAEM introduces a learnable frequency-domain filtering pipeline operating on the Fourier magnitude spectrum while preserving phase information. It selectively amplifies high-frequency textural signatures of small defects (e.g., cracks and pollution flashovers) often missed by conventional detectors (YOLOv5n, YOLOv8s, and LiteYOLO-ID (2024)). The proposed approach was evaluated on three public datasets: IDID-Plus (~1,600 images), PLIDD (~4,927 images), and SFID (~320 images). The IDID-Plus and PLIDD datasets were used for training, validation, and testing with a 70/15/15 split, while SFID was employed exclusively for benchmarking robustness under degraded visibility. The performance was measured using Mean Average Precision (mAP) derived from the area under the precision recall curve. All experiments were conducted in a reproducible Google Colab Pro environment using an NVIDIA Tesla T4 GPU and PyTorch 2.6.0. The experimental results demonstrated that FAEM-YOLO achieved 90.7% mAP@0.5 and 50.2% mAP@0.5:0.95 on IDID-Plus, significantly outperforming LiteYOLO-ID (2024). On PLIDD, the model attained 73.9% mAP@0.5 and 40.5% mAP@0.5:0.95, surpassing the YOLOv8n baseline by 3.9 points on the stricter metric while using 20% fewer parameters (4.2M vs. 5.3M). Furthermore, FAEM-YOLO recorded 99.2% mAP@0.5 and 84.4% mAP@0.5:0.95 on the SFID dataset. These

results underscore that frequency-domain enrichment generalizes well on different dataset; hence, it is an efficient strategy for visual inspection in UAV-based power line monitoring.

Keywords: tiny object detection; UAV inspection; insulator defects; lightweight deep learning; frequency-domain features; Fast Fourier Transform (FFT); computer vision

1. Introduction

Electrical power transmission networks are a very important infrastructure for reliability and influences economic stability and development [1,2]. For sub-Saharan Africa, especially Ghana, a stable power supply is paramount; minor failures in High Voltage transmissions lines can set off a widespread blackout, which can significantly cause economic losses and endanger lives and properties [3]. To ensure undisrupted power supply, frequent maintenance and inspections of the transmission line components, such as conductors, insulators, towers, and fittings, are required. Traditionally, power line inspections have been performed via ground patrols or helicopter surveys. Ground crews on foot can sometimes spot local issues, but this method is extremely time-consuming, labor-intensive, and limited in coverage [4]. Helicopter surveys cover broader areas faster, yet they entail high operational costs, dependence on weather conditions, and notable safety risks to crews [4,5]. This need has motivated the exploration of more advanced and automated inspection methods to enhance grid reliability.

In particular, insulator defect detection has become a focal task in this domain. Insulators are critical devices that electrically isolate live wires from the supporting towers (preventing leakage currents) and are commonly made of ceramic, glass, or polymer materials [6]. Over time, insulators may develop subtle defects (e.g., hairline cracks, flashover marks, or contamination build-up) that often manifest as tiny textural anomalies. Such small faults are difficult to spot with the naked eye or standard visual inspection from a long distance; yet, if left undetected, they can precipitate flashovers or failures that bring down an entire line [7]. Therefore, early and accurate detections of insulator defects are crucial to prevent major outages and accidents [8]. These limitations have spurred interest in leveraging new technologies to improve grid monitoring and inspection.

In recent years, unmanned aerial vehicles (UAVs) equipped with high-resolution cameras have emerged as a promising alternative for power line inspection [9]. Drones provide an agile, scalable, and safer approach: They can survey vast stretches of line, including hard-to-reach or hazardous terrain, without putting human crews at risk [4,9,10]. However, a drone captures only data, as the real breakthrough comes from pairing UAVs with intelligent computer vision. By integrating onboard or off-board deep learning algorithms, UAVs can automatically analyze imagery in real time to detect faults or anomalies, rather than relying on offline review of footage [10–12]. These models are typically large and computationally intensive. This makes direct deployment on drones difficult, as UAVs carry lightweight computers and have strict limits on weight, power, and processing capacity [13,14].

To address this efficiency issue, Ghost Convolution and related model compression techniques are used to reduce model size and computation [15]. However, this modification often sacrifices detection accuracy, especially on very small or subtle targets [16]. In the context of insulator inspection, most compact detectors struggle to capture the fine textural cues of tiny defects,

essentially trading precision for speed [17,18]. This trade-off is unacceptable in critical grid monitoring since missed detections of "small" insulator faults could lead to catastrophic failures [19].

Furthermore, modern object detectors (e.g., the You only look once (YOLO) family or transformer-based models) achieve high accuracy in general visual tasks [11,20–22], including powerline component detection. In reference [14], GCL-YOLO used GhostConv layers and a small-object oriented detection head to achieve large parameter reductions (>70%) with competitive accuracy on UAV benchmarks. In similar work [23], ID-YOLO applied a task-specific Lightweight Channel-Spatial Attention (LCSA) module and GhostNet backbone on embedded UAV hardware for insulator defects. The results showed improved recall confirming balance sensitivity against false alarms. In another study [24], LiteYOLO-ID combined an Efficient Channel Attention-GhostNet-C2f (EGC) block and IDID-Plus dataset to improve accuracy while significantly reducing parameters. Similarly, the researchers in [25] introduced an additional high-resolution detection layer and lightweight attention to enhance small-object sensitivity in YOLO-TLA. Beyond pure model compression, the researchers in [26] introduced a High-Frequency Perception (HFP) module and a Spatial Dependency Perception (SDP) module to amplify high-frequency features and preserve spatial alignment for tiny objects. The performance demonstrated superior recall and precision on dedicated tiny-object datasets. Despite the effectiveness of the reviewed techniques, they have limitations of either relying on fixed high-pass filter or spatial-domain attention, leaving learnable frequency-domain enrichment underexplored for capturing subtle, defect-specific textural cues [27–29].

To conclude, the central problem addressed in this study is the lack of a vision-based detection framework that can deliver high accuracy identification of tiny insulator defects in UAV power line inspections. High-frequency image information (edges, abrupt texture changes) often corresponds to the subtle details of small defects that conventional neural network (CNN) features might overlook. It is hypothesized that integrating a Frequency-Aware Enrichment Module (FAEM) into a YOLO-based network amplifies critical high-frequency cues, improving the detector's ability to "see" tiny insulator cracks or contaminants. This raises the central research question: Can frequency-domain feature enrichment enable a lightweight, UAV-deployable detector to achieve accuracy on tiny insulator defects without compromising real-time performance? To answer the question, we explore a new approach that augments a one-stage detector with frequency-domain feature analysis in order to boost sensitivity to fine defects without bloating the model. The proposed FAEM-YOLO introduces a learnable frequency-domain filtering mechanism that is optimized end-to-end within the detection network. FAEMs adaptively emphasize or suppress frequency components most relevant to defect characteristics during training. By integrating this adaptive spectral enrichment into a lightweight YOLO architecture, the proposed method establishes an accuracy balance for insulator defect detection. Addressing this challenge has practical importance and scientific value. Practically, improving defect detection translates to a more resilient grid with fewer unexpected outages, which can save utility companies significant costs on downtime and manual inspections [4,29] and aligns with ongoing efforts to strengthen power infrastructure reliability (as in Ghana's energy sector initiatives [3,30]).

2. Materials and methods

2.1. Theoretical background

2.1.1. Lightweight convolutional models and ghost convolution

Real-time object detection on resource-constrained platforms, such as UAVs, requires compact yet expressive neural networks. Conventional CNNs employ convolutional layers with weight tensors of size,

$$C_{in} \times C_{out} \times k \times k \tag{1}$$

where C_{in} and C_{out} are the number of input and output channels, and $\boldsymbol{k}$ is the kernel size. The number of parameters is therefore approximately

$$Parameters \sim C_{in} \cdot C_{out} \cdot k^2 \tag{2}$$

and the computational cost for a feature map of size $H \times W$ is

$$O(C_{in} \cdot C_{out} \cdot k^2 \cdot H \cdot W) \tag{3}$$

This quickly becomes expensive, especially for real-time detection tasks on UAVs, where hardware resources are limited. Moreover, many of the generated feature maps are redundant or highly correlated.

To address this inefficiency, GhostNet Convolution [15,16,31] was introduced. The core idea is to generate only a small set of intrinsic features using standard convolutions and then expand them into a larger set of ghost features using cheap linear operations (e.g., depthwise convolution). As illustrated in Figure 1, the input feature map is first processed by a conventional convolution to produce intrinsic features, after which multiple inexpensive linear transformations are applied to generate ghost feature maps. These intrinsic and ghost features are subsequently concatenated to form the final output representation, significantly reducing computational cost while preserving representational capacity.

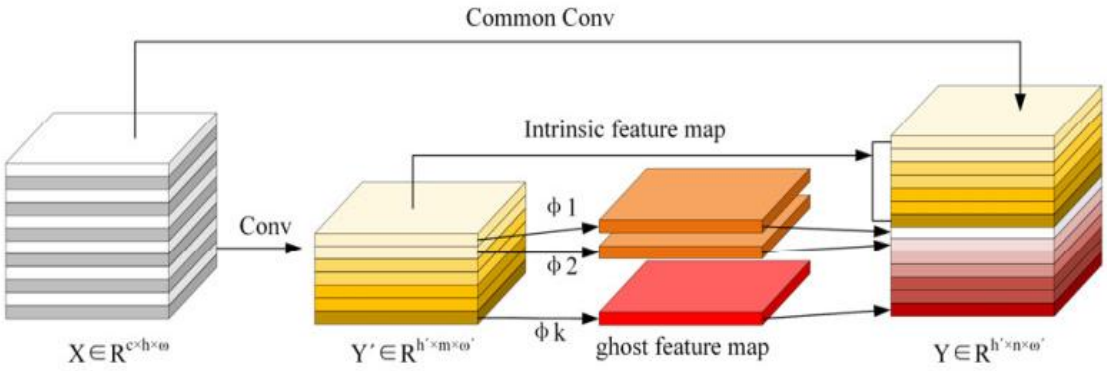

Figure 1. Overview of the ghost convolution module [15].

Formally, for an input tensor:

$$X \in \mathbb{R}^{B \times Cin \times H \times W} \tag{4}$$

where $\boldsymbol{B}$ is the batch size. A Ghost module produces an intermediate intrinsic feature map

$$Y_{intrin\ sic} = \Phi_{conv}(X), \qquad Y_{intrinsic} \in \mathbb{R}^{B\times C_p\times H'\times W'} \tag{5}$$

With $\boldsymbol{C_p} = \frac{C_{out}}{2}$. The remaining $\boldsymbol{C_p}$ channels are generated as ghost features via cheap operations

$$Y_{ghost} = \Phi_{\text{cheap}}(Y_{intrinsic}) \tag{6}$$

and the final output is the concatenation of the two streams;

$$Y_{out} = Concat\left[Y_{intrinsic}, Y_{ghost}\right] \in \mathbb{R}^{B\times C_p\times H'\times W'} \tag{7}$$

This design drastically reduces FLOPs and parameters while retaining accuracy. Ghost modules have since been integrated into lightweight detectors such as GCL-YOLO, LiteYOLO-ID, and GhostNetV3, demonstrating competitive accuracy at a fraction of the cost.

However, aggressive compression comes with tradeoffs. By relying heavily on cheap operations, Ghost modules can lose sensitivity to very fine details, which are crucial for tasks like tiny insulator defect detection. Subtle cracks or pollution marks often manifest as high-frequency textures, and lightweight CNNs operating only in the spatial domain may fail to capture them.

2.1.2. Fast Fourier Transform (FFT) and frequency-domain analysis

The Discrete Fourier Transform (DFT) decomposes a spatial feature map $f(x,y)$ into its spectral components

Formally, the 2D Discrete Fourier Transform (DFT) of a feature map $f(x,y)$ of size $H\times W$ is:

$$F(u,v) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y)e^{-j2\pi\left(\frac{ux}{H}+\frac{vy}{W}\right)} \tag{8}$$

where $F(u,v)$ is a complex spectrum consisting of magnitude $|F(u,v)|$ and phase $\angle F(u,v)$. The original feature map can be exactly recovered through the inverse transform;

$$f(x,y) = \mathcal{F}^{-1}\{F(u,v)\} \tag{9}$$

In practice, the FFT computes the DFT with far lower cost, reducing complexity from $O(N^2)$ to $O(N\ log\ N)$. For a 2D feature map of size $H\times W$, the cost is $O(HW\ log\ (HW))$, which is efficient with modern implementations (e.g., FFTW and cuFFT). In image analysis, low-frequency coefficients (near the spectrum center) correspond to smooth global structures, while high frequencies (at the edges) capture fine details such as edges, cracks, or flashover marks [32–34]. By the Convolution Theorem,

$$\mathcal{F}\{f * g\} = F\cdot G \tag{10}$$

The Fourier transform offers a global representation of feature maps, where convolution becomes simple multiplication in the frequency domain. This global view enables the capture of subtle, high-frequency patterns such as fine cracks or pollution marks that are often overlooked by CNNs, which tend to favor low and mid-frequency information [35]. By embedding FFT into the network

pipeline, these high-frequency cues are directly enhanced, improving sensitivity to small defects without adding heavy computational cost. Unlike earlier uses of Fourier-related transforms, such as the DCT in compression (e.g., JPEG), our approach leverages FFT, not for compression, but as a feature enrichment mechanism directly augmenting spatial features with frequency-domain information to improve insulator defect detection [33,36].

2.1.3. Evaluation metrics

In object detection, performance is commonly measured using Mean Average Precision (mAP), derived from the area under the precision–recall curve. The VOC criterion (mAP@0.5) considers detections correct at IoU ≥ 0.5, while the COCO metric (mAP@0.5:0.95) averages over stricter IoU thresholds (0.5–0.95), offering a tougher assessment, especially for small objects.

$$AP = \int_0^1 P(R)dR \quad , \quad mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{11}$$

Complementary metrics include Precision (P), the proportion of correct detections among predictions, and Recall (R), the proportion of true objects successfully detected. Together, they balance sensitivity against false alarms.

$$Precision = \frac{TP}{TP + FP} \quad , \quad Recall = \frac{TP}{TP + FN} \tag{12}$$

Efficiency is equally important for lightweight models and is assessed by model size (parameters) and computational complexity (GFLOPs). Finally, qualitative analysis of detection outputs provides additional insight into robustness under challenging conditions beyond numerical scores.

Datasets

We evaluated the method on three datasets: IDID-Plus, PLIDD, and SFID. IDID-Plus contains ~1,600 high-resolution images with over 5,300 annotated instances across three class insulators, pollution-flashover defects, and broken segments captured under varied real-world conditions, and split into 1,296/144/160 train, val, and test sets. PLIDD was created by us to further assess generalization. PLIDD comprises 4,927 images compiled from multiple sources. In particular, we incorporated a public Chinese insulator-defect image dataset (CPLID) with high-voltage line insulators (ranging from ~132 kV to 330 kV) [37] as a base, and we augmented it with additional images and backgrounds from the IDID dataset for greater diversity, and split 70/15/15, shown in Table 1, using Roboflow for the dataset preparation. Representative samples and corresponding annotations from the PLIDD dataset are illustrated in Figure 2. SFID augments insulator images with synthetic fog (50% foggy, 50% clear) using atmospheric and noise-based models, providing paired samples to assess robustness under degraded visibility.

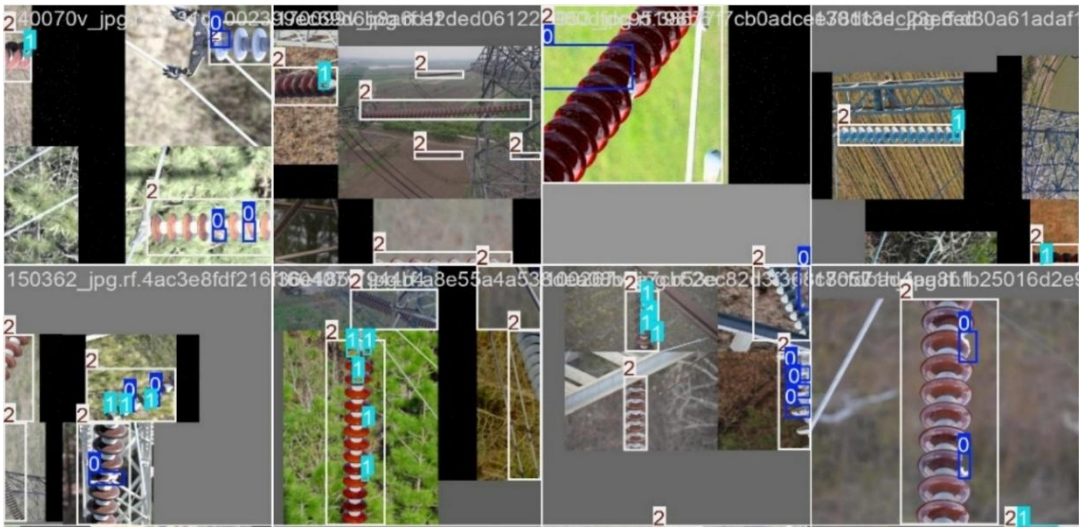

Figure 2. Sample image and annotation from the PLIDD dataset.

Table 1. Overview of datasets, their splits, and roles in training or evaluation.

Dataset	Train/Val/Test	Total Images	Total ID Instances	Notes
IDID-Plus	1296 / 144 / 160	~1,600	5,300+	Public dataset real images (outdoor)
PLIDD	3449 / 739 / 739	4,927	7,200+ (approx.)	Compiled from IDID- plus and CPLID; varied scenes
SFID	– / – / 320	320	800 (paired 1:1 clear/fog)	Synthetic fog applied to test images; evaluation only

2.2. Preprocessing

All images were resized to 640×640 pixels for training and inference. We applied extensive data augmentation during training to improve generalization. Each training image was randomly subjected to geometric transforms (horizontal flip with 50% chance; random rotation up to ~5–10°; scaling/zooming up to ±10%; translation up to ±10% of image size) and photometric adjustments (random brightness and contrast jitter within ±20% range). We also employed the mosaic augmentation (combining 4 images into one) as used in YOLOv8 for 70% of training epochs, which helped expose the model to varied context and small object placements.

Additionally, to address class imbalance in PLIDD (fewer broken insulator examples), we used targeted copy-paste augmentation: Some "broken insulator" defect regions were segmented and pasted onto other backgrounds to increase their frequency in training. No augmentations were applied to validation or test images beyond resizing. The heavy augmentation regimen (referred to as our “high-augmentation” training) was applied in our later training runs to push the model to higher performance. We differentiated this from a “baseline” training run with only standard augmentations (flips, minor photometric changes, but no mosaic or copy-paste), as discussed in the results.

Furthermore, it is important to clarify the training strategy across datasets. We trained and evaluated models on each dataset separately to avoid any data leakage or overly optimistic results. For our ablation studies and initial development, we trained on the PLIDD training set (to take

advantage of its diversity) and evaluated on the PLIDD test. For the final benchmarking against other methods, we trained a model from scratch on the IDID-Plus training set and evaluated it on the IDID-Plus test set, so that comparisons with published results were fair. The SFID foggy dataset was used only to test the already-trained IDID-Plus model's performance under fog; i.e., the model did not see foggy images during training.

2.3. Network architecture

Figure 3 depicts the proposed FAEM-YOLO architecture. The input image was first processed by a YOLOv8-style backbone built with GhostConv modules, where each GhostConv layer convolved only half its channels and then concatenated these convolved features with the remaining channels. We inserted Frequency-Aware Enrichment Modules (FAEMs) into this backbone: One immediately after the first downsampling stage (P2, at 1/2 the input scale) and another at a mid-level stage (P4, at 1/8 scale) and so on. Each FAEM (zoomed in on the right of Figure 3) split its input into two parallel streams. One stream processed features in the spatial domain as usual, while the other stream operated in a frequency domain using a learnable spectral mask.

The spectral-processed and spatial-processed features were then concatenated channel-wise to form an enriched feature map. This dual-stream design embedded frequency-aware filtering into the network, effectively integrating spatial and spectral information before the features continued through the remaining backbone layers and the YOLO detection head. The enriched multi-scale features were then fed to the standard YOLO prediction head to produce the final bounding-box defect detections.

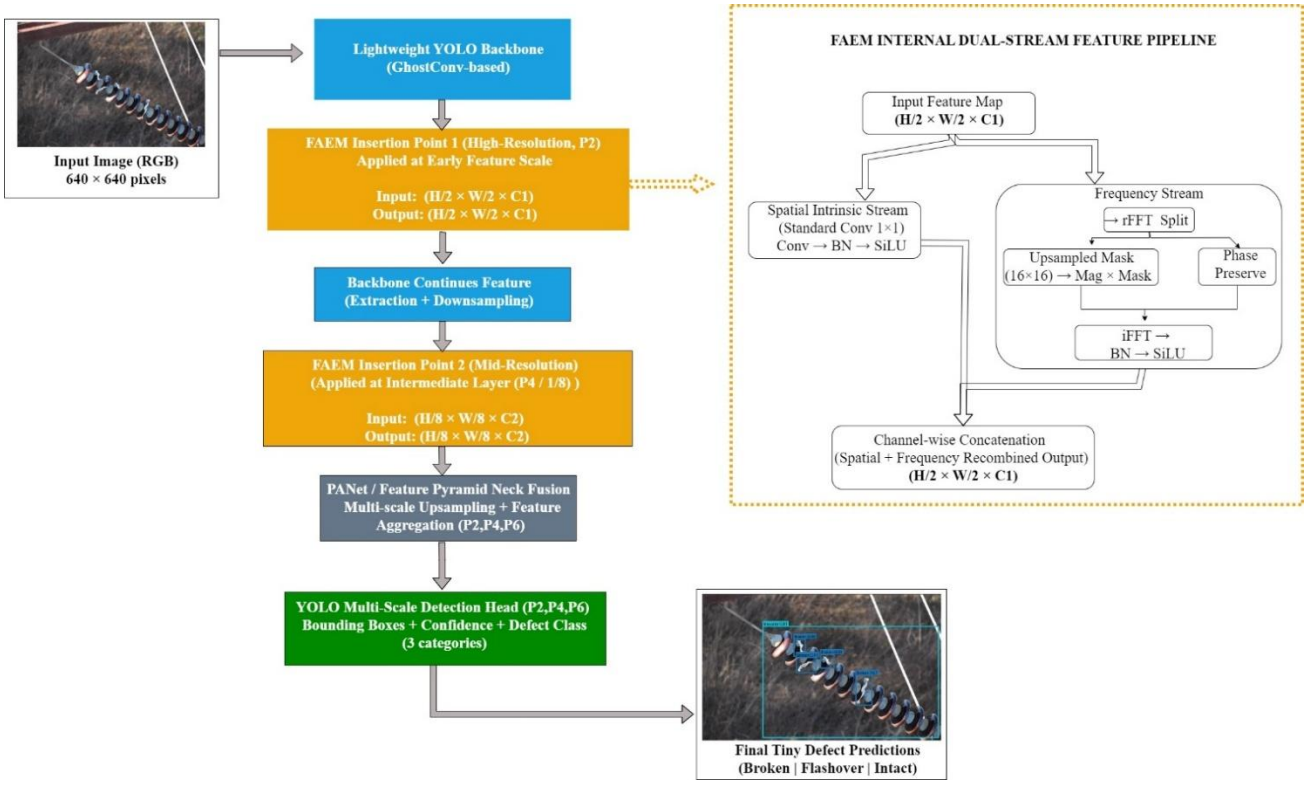

Figure 3. Overview of the Frequency-Aware Enrichment Module (FAEM) integrated into a lightweight YOLO architecture for tiny insulator defect detection.

The key novelty in the architecture was the insertion of FAEM at strategic points in the feature hierarchy (highlighted in orange in Figure 3). We placed an FAEM block after the initial layers at 1/2 scale (P2) and another in the middle of the backbone at 1/8 scale (P3). The rationale for these choices was as follows: The 1/2-scale feature map (after the first downsampling) was the highest-resolution feature with fine-grained information; by enriching it with frequency-domain features early on, we ensured that even the tiniest defect patterns were amplified from the beginning of the feature extraction process. The 1/8-scale feature (around the middle of the backbone) represented a compromise between resolution and abstraction, where it was coarse enough to incorporate some context and larger receptive fields, yet fine enough to localize small defects.

Inserting FAEM here enabled the network to reinforce high-frequency details at an intermediate level, before the features were distributed to the detection head. We empirically found that these two points offered the best boost in performance: Placing FAEM only at the very early stage might not have propagated frequency cues through deeper layers effectively, while adding it at multiple scales ensures low-level textures and mid-level features were enhanced. We opted not to insert FAEM at every scale to avoid excessive overhead; instead, targeting two scales provided a good balance of accuracy gain vs. complexity. The ablation studies confirmed that using two FAEM blocks (at 1/2 and 1/8 scales) yielded significant improvements, whereas additional blocks gave diminishing returns. With the FAEM blocks in place, the network's forward flow was as follows (refer to Figure 3 for a high-level workflow and Figure 4 for architecture details): An input image first passed through a stem convolution and downsampling. At this point (1/2 scale feature), the first FAEM module processed the feature map, injecting frequency-enhanced details. The backbone then continued with GhostConv and CSP layers, downsampling to 1/4 and then to 1/8 scale, where the second FAEM module was applied.

This ensured that by the time features reached deeper layers, the fine textures and some mid-level patterns were frequency-enriched. The neck then upsampled and fused features from 1/8 (with FAEM), 1/16, and 1/32 scales. Finally, the YOLO detection head outputted bounding boxes and class predictions at 1/8, 1/16, and 1/32 scales (P2, P4, P6), which corresponded to detecting small, medium, and large insulator objects, respectively. Because tiny defects mainly appeared on the insulators (which can be small objects in the image), enhancing P3 (the finest prediction scale) via FAEM was especially beneficial. Overall, this architecture was designed to progressively enrich features with frequency information without significantly increasing model size. The final model had ~4.2 million parameters and ~15.8 GFLOPs, which was smaller than the baseline YOLOv8n (5.3M, 17.2 GFLOPs) due to our GhostConv compression; yet, it achieved much higher accuracy thanks to FAEM.

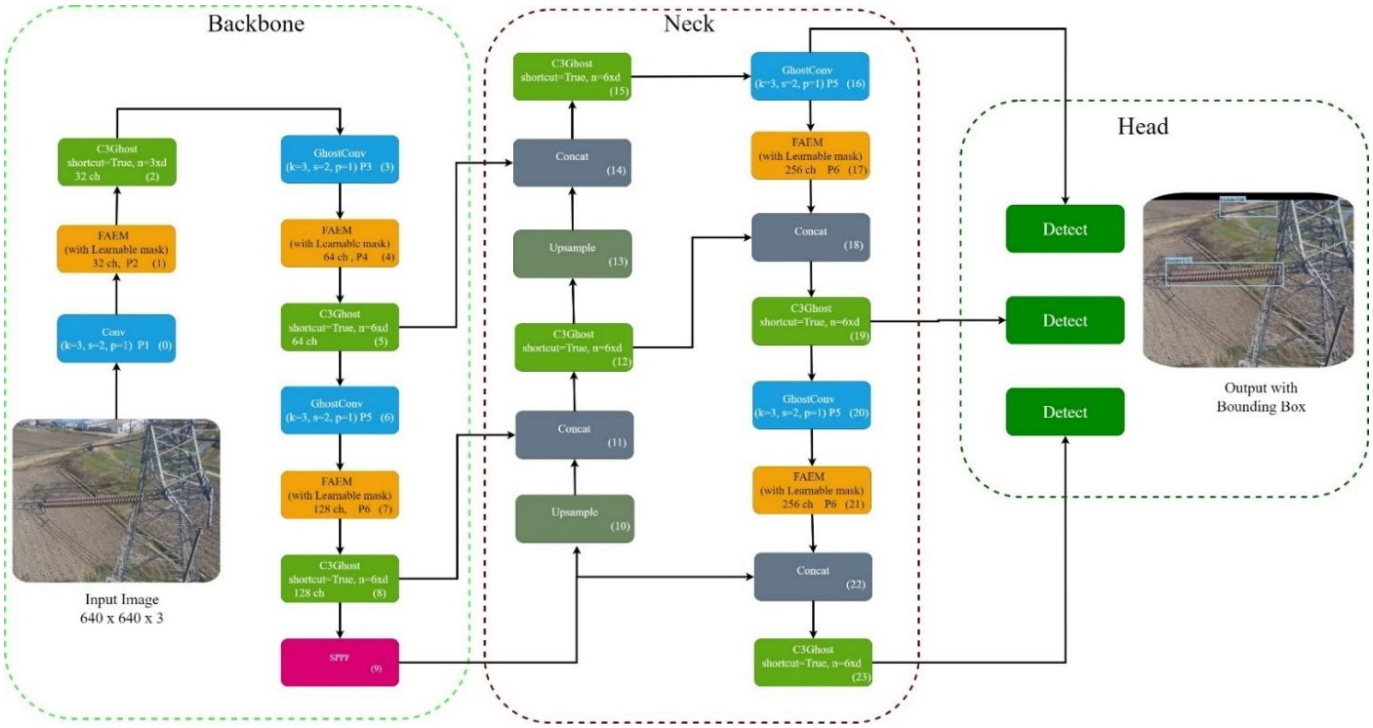

Figure 4. FAEM-YOLO detailed network architecture.

2.4. Proposed frequency-aware enrichment module

Building upon the concepts of Ghost Convolution (Eq. 1-7) and FFT-based frequency-domain analysis (Eq. 8-10), we proposed the FAEM. The goal was to overcome the limitations of aggressive compression in lightweight models (Eq. 2–3), which often fail to capture subtle high-frequency anomalies such as cracks or flashover marks. FAEM introduces a dual-stream design; a spatial stream derived from standard convolution and a frequency-aware stream that enhances fine textures via learnable spectral filtering, as shown in Figure 5.

To enhance clarity on implementation, the input feature map first passed through a primary 1×1 convolution layer that reduced its channel dimension by half (e.g., 256 → 128). The spatial path retained local texture details using standard 3×3 convolutions (output shape $H \times W \times 64$). In parallel, the frequency-aware path applied a real FFT (rFFT), generating a complex tensor split into magnitude and phase (H×W×C/2). After learnable masking, normalization, and inverse rFFT, the resulting frequency-refined features $(H \times W \times 64)$ were concatenated with spatial features to form the output $(H \times W \times 128)$ used in downstream detection.

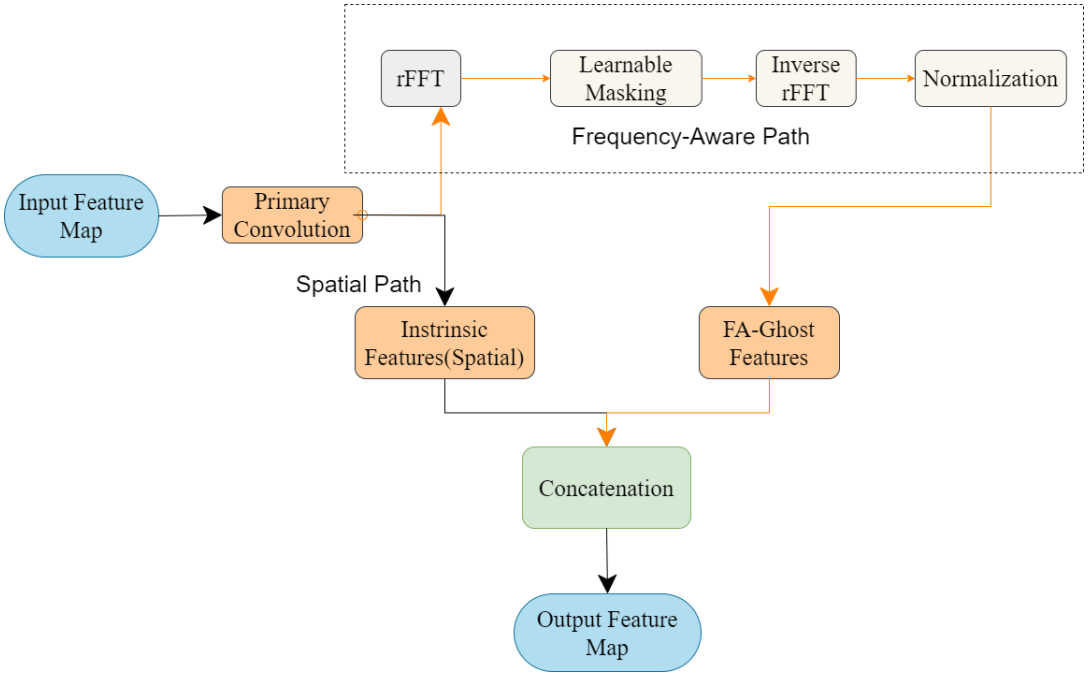

Figure 5. Internal diagram of the Frequency-Aware Enrichment Module (FAEM).

Frequency-Aware Path (Learnable Magnitude–Phase Masking)

The detailed frequency-aware path enrichment process within the FAEM is illustrated in Figure 6, building upon the module overview shown in Figure 5. The FFT output was split into magnitude and phase. A learnable mask selectively modulated the magnitude while the phase remained unchanged. The modified spectrum was then recombined and passed through an inverse FFT to produce frequency-enhanced features.

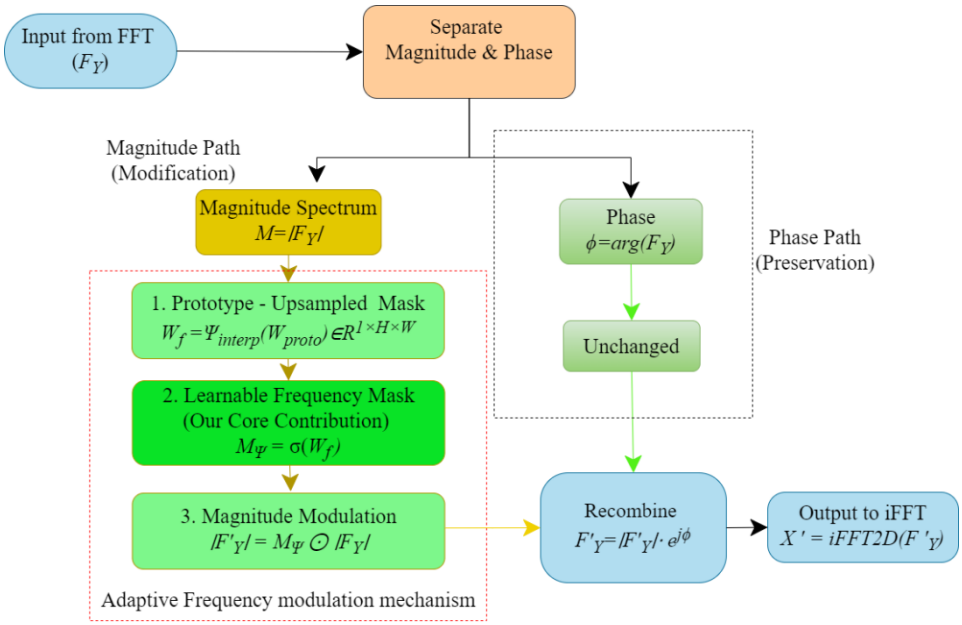

Figure 6. Learnable magnitude phase masking mechanism within the FAEM.

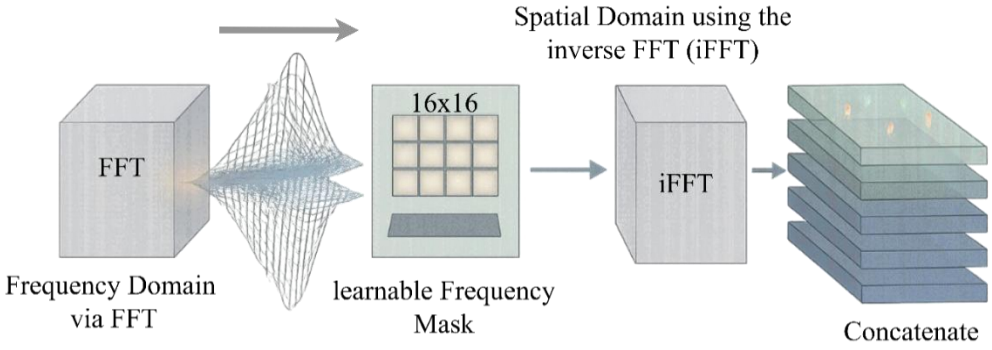

Figure 7. Schematic of the FAEM. The input feature map is transformed into the frequency domain via FFT, where a learnable 16×16 frequency mask selectively amplifies or suppresses spectral components.

As outlined in Figures 6 and 7, FAEM operated on the complex spectrum produced by the Real Fast Fourier Transform (Eq. 8). The spectrum was decomposed into its magnitude ($|F_Y|$) and phase ($\angle\mathcal{F}_Y$). A compact learnable prototype mask (W_{proto}) was introduced, which was upsampled to the resolution of the magnitude spectrum. By element-wise multiplication, the mask selectively suppressed or amplified certain frequency components, yielding a filtered magnitude (M'). The phase was preserved to ensure correct spatial alignment.

Input and Output Definitions

Let the input tensor be:

$$X \in \mathbb{R}^{B\times Cin\times H\times W} \quad , \tag{13}$$

where B is the batch size, C_{in} is the input channel, and $H \times W$ is the spatial resolution. The output has the form:

$$Y_{out} \in \mathbb{R}^{B\times C_{out}\times H'\times W'} \quad . \tag{14}$$

where (H', W') depend on stride and padding, which is consistent with the standard convolutional mapping in Eq. 5-7.

Spatial Stream (Intrinsic Features)

The first half of the output channel $\boldsymbol{C_p = \frac{C_{out}}{2}}$ is produced using a primary convolutional block Φ_{conv}, analogous to Eq. 5:

$$Y_{intrinsic} = \Phi_{conv}(X), \qquad Y_{intrinsic} \in \mathbb{R}^{B\times C_p\times H'\times W'} \tag{15}$$

This stream captures edges, shapes, and contextual object structure, similar to the intrinsic pathway of Ghost Convolution (Eq. 6).

Frequency-Aware Feature Generation (Ghost Stream)

In contrast to traditional Ghost modules that use cheap linear transformations [15] (depthwise convolutions), our FAEM introduced a learnable frequency-domain pathway to capture textural richness and defect-oriented patterns. This stream generated the remaining "ghost" feature maps in 5 steps, which took the just-computed intrinsic features (see Appendix A for the full mathematical formulation of each step).

Step 1: Transformation to Frequency Domain

The intrinsic feature map was projected into the frequency domain via a 2D Real FFT (extending Eq.8): This produced a complex frequency spectrum representing the feature map's content in terms of magnitudes and phases across frequencies (see Eq. 16 in Appendix A).

Step 2: Learnable Frequency Filtering

At the heart of the FAEM module lies its key innovation, a learnable frequency-domain filtering mechanism. A compact prototype mask W_{proto} was defined as a learnable 16×16 matrix (see Eq. 17 in Appendix A), which served as a compact frequency filter template. This prototype mask was then upsampled to the full FFT size by an interpolation function Ψ_{interp} (see Eq. 18 in Appendix A), producing a full-resolution frequency mask. The upscaled mask was applied to the FFT's magnitude spectrum via element-wise multiplication (see Eq. 19 in Appendix A), selectively scaling each frequency component. Moreover, the original phase spectrum was preserved and recombined with the modified magnitude (see Eq. 20 in Appendix A) to form a filtered complex spectrum (a sigmoid normalization was applied to the mask in this process to constrain its values within a bounded range, as formalized in Eq. 21 in Appendix A). This learnable frequency masking operator denoted ($\boldsymbol{\mathcal{M}_{\Psi}}$) **"*our core contribution*"** enabled the network to amplify frequencies indicative of defects (e.g., the high-frequency signals of hairline cracks or fine pollution textures) and suppress irrelevant noise. Through training, W_{proto} learned to emphasize those frequency components most relevant to insulator defect detection. Notably, because W_{proto} was extremely small, 16×16, and then interpolated to full size, it introduced only a minimal computational overhead. This design provided powerful feature enrichment at low cost, aligning with the efficiency requirements of lightweight UAV-based inspection models.

Step 3: Transformation Back to the Spatial Domain

The enriched spectrum was transformed back into the spatial domain using the inverse FFT denoted $\left(\boldsymbol{\mathcal{F}_{irfft2}^{-1}}\right)$. This yielded a preliminary enriched feature map in the spatial domain, wherein the previously emphasized frequency components appeared as enhanced spatial features. In essence, the subtle defect cues amplified in the frequency domain (from Step 2) were reintroduced as spatial patterns that the CNN could further process (see Eq. 22 in Appendix A).

Step 4: Normalization

The reconstructed ghost feature map was then normalized to stabilize learning. Batch Normalization was applied followed by a SiLU activation (a sigmoid-weighted linear unit) to introduce nonlinearity (see Eq. 23 in Appendix A). This normalization step ensured numerical stability and appropriately scaled the enriched features before they were fused with the original features, preventing any extreme frequency responses from disrupting the training process.

Step 5: Final Output Generation

Finally, the intrinsic (Eq. 15) and frequency-aware features (Eq. 23) were concatenated channel-wise:

$$Y_{out} = Concat\left[Y_{intrinsic}, Y_{ghost}\right], along\ dim = 1, \tag{24}$$

producing the hybrid tensor:

$$Y_{out} \in \mathbb{R}^{B \times C_{out} \times H' \times W'}, \tag{25}$$

This design mirrored the GhostNet principle of dual feature streams (Eq. 7) but replaced linear cheap operations with FFT-based spectral enrichment, ensuring sensitivity to subtle anomalies while retaining computational efficiency.

$$Y_{out} = \text{Concat}\left[\underbrace{\Phi_{conv}(X)}_{\text{Spatial Stream}}, \underbrace{\sigma\left(BN\left(\mathcal{F}_{irfft2}^{-1}\left(M_{\Psi}(\mathcal{F}_{rfft2}(\Phi_{conv}(X)))\right)\right)\right)}_{\text{Frequency-Aware Stream}}\right] \tag{26}$$

Φ conv (·)	Standard convolutional block
$\mathcal{F}_{rfft\,2}$	Real-valued 2D Fast Fourier Transform
$\boldsymbol{\mathcal{F}_{irfft2}^{-1}}$	Inverse of real 2D FFT
$\boldsymbol{\mathcal{M}_{\Psi}}$	The Frequency Masking Operator (Our Contribution)
σ	Non-linear activation function (SiLU)
BN	Batch Normalization
Concat	Channel-wise concatenation

2.5. Training setup and implementation

All experiments were conducted within a consistent and reproducible Google Colab Pro environment, utilizing an NVIDIA Tesla T4 GPU with 16 GB of VRAM and a PyTorch 2.6.0 framework. Our implementation was based on the Ultralytics YOLOv8 (v8.3.179) high-level API, which was extended to support our custom architectural modules. These modules, including the proposed FAGhostModule, were dynamically registered at runtime to enable model construction from our bespoke YAML configuration files. The PLIDD (v6), sourced from Roboflow, served as the primary dataset for our ablation studies, with images cached in RAM to ensure efficient data loading.

All the models were trained under an identical, high-performance protocol. We employed an AdamW optimizer with a weight decay of 0.0005 and a momentum of 0.937. The training was conducted with a batch size of 16 for 200 epochs, using a cosine annealing learning rate scheduler with an initial learning rate of 0.01 and a 3-epoch warm-up. Early stopping was configured with a patience of 50 epochs based on the validation mAP@0.5:0.95 metric to select the best-performing model checkpoint.

A comprehensive data augmentation strategy was enabled, including mosaic augmentation for the first 140 epochs, to enhance model generalization. The training objective utilized the standard YOLOv8 composite loss function. To ensure a fair and direct comparison of architectural merit, all

models in our study were trained from scratch, with randomly initialized weights, rather than using checkpoints pretrained on external datasets like COCO. This approach guaranteed that all learned features were specific to the insulator defect detection task.

3. Results

Presented are the experimental results of the developed FAEM-YOLO model on the insulator defect detection task. We reported the performance under different training scenarios (baseline vs high-augmentation) and compared the model's accuracy and speed against baseline models. An ablation analysis was conducted to quantify the contribution of the FAEM module and related improvements. The results were summarized in tables and figures, followed by interpretation of what they mean for the research questions. Additionally, examination of some qualitative outcomes was conducted (visualizations of detections and internal feature behaviors) to gain deeper insight into how the model was working.

Performance under Baseline and Enhanced Training Regimes

After Run 1 (baseline training), the FAEM-YOLO model demonstrated strong performance on the PLIDD dataset. Without heavy augmentation, it achieved a mean Average Precision (mAP) @ 0.5 IoU of around 70.3%, and an overall Recall of about 65%. This confirmed that the architecture was fundamentally sound, the model successfully learned to detect the three defect classes, outperforming our baseline YOLO (which achieved roughly 67% mAP under the same conditions).

However, there was room for improvement in sensitivity (recall). The training was subsequently extended to Run 2 (high-augmentation), which yielded a model with significantly improved performance across all metrics. Impact of Data Augmentation, the high-augmentation training (Run 2), yielded substantial improvements in detection accuracy compared to Run 1. Table 2 summarizes the key metrics from the two runs.

Table 2. Baseline vs heavy-augmentation FAEM-YOLO: Accuracy (mAPs), recall/F1, missed flashovers %, and background false alarms.

Metric	Run 1 (Baseline)	Run 2 (High-Aug.)	Improvement
Peak mAP @ 0.5	70.3%	73.9%	+3.6 points
Peak mAP @ 0.5:0.95	38.0%	40.6%	+2.6 points
Overall Recall	65%	70%	+5.0 points
Optimal F1 Score	0.72	0.75	+0.03
Missed Flashover Defects	42%	34%	-8.0 points
Background False Positives	307	247	-19.5%

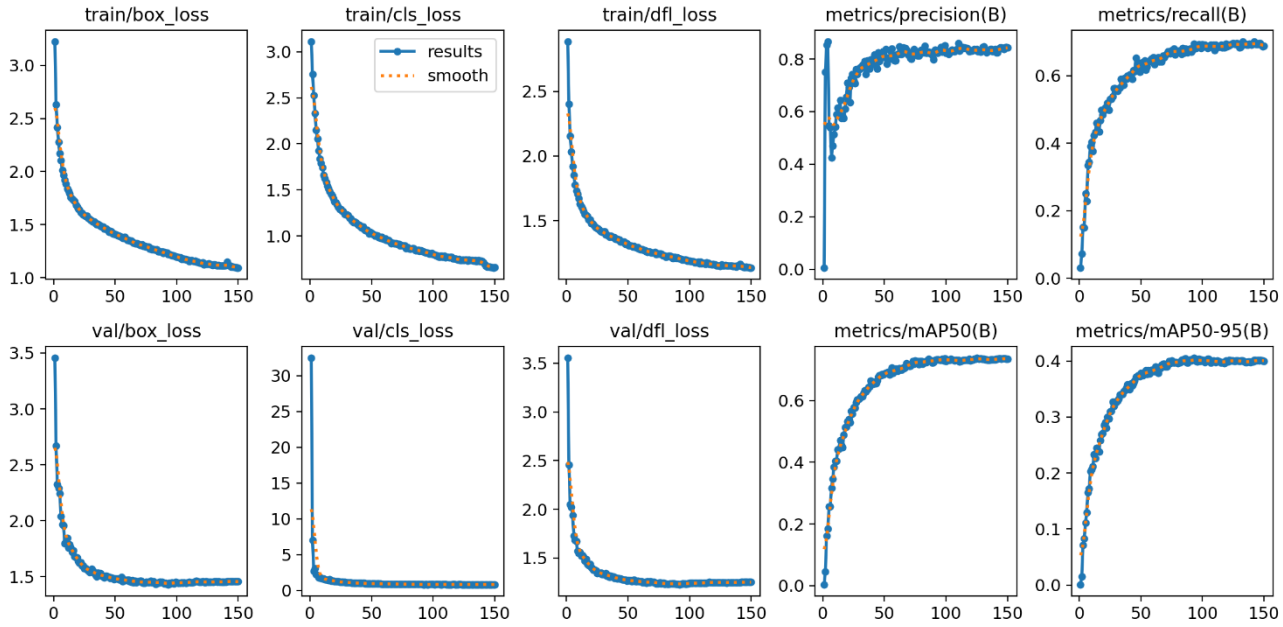

Figure 8. Run 1 (Baseline) training metrics over the course of training.

Figure 8 presents the training metrics over the course of training. In Run 1, the loss curve and accuracy stabilized by around 100 epochs (Figure 8), indicating convergence. Training beyond this point yielded diminishing returns, so Run 1 was conducted for only 150 epochs. all three training losses (box, classification, and DFL) decreased monotonically over the epochs, indicating that the model was learning effectively. The validation losses followed a similar downward trend, flattening out toward the end of training, which suggested that the model was not overfitting severely to the training data.

Concurrently, the model's accuracy metrics improved over time: Precision and recall on the validation set started lower but rose throughout training, and the mAP curves (mAP@0.5 and mAP@0.5:0.95) steadily increased. By the end of Run 1, the FAEM-YOLO reached approximately 70% mAP@0.5 and 38% mAP@0.5:0.95, with a final recall around 65%. This baseline training run established a solid performance level, but the plateauing of recall and mAP in Figure 8 indicated that the model may have reached the limits of what it could learn from the dataset with standard augmentation.

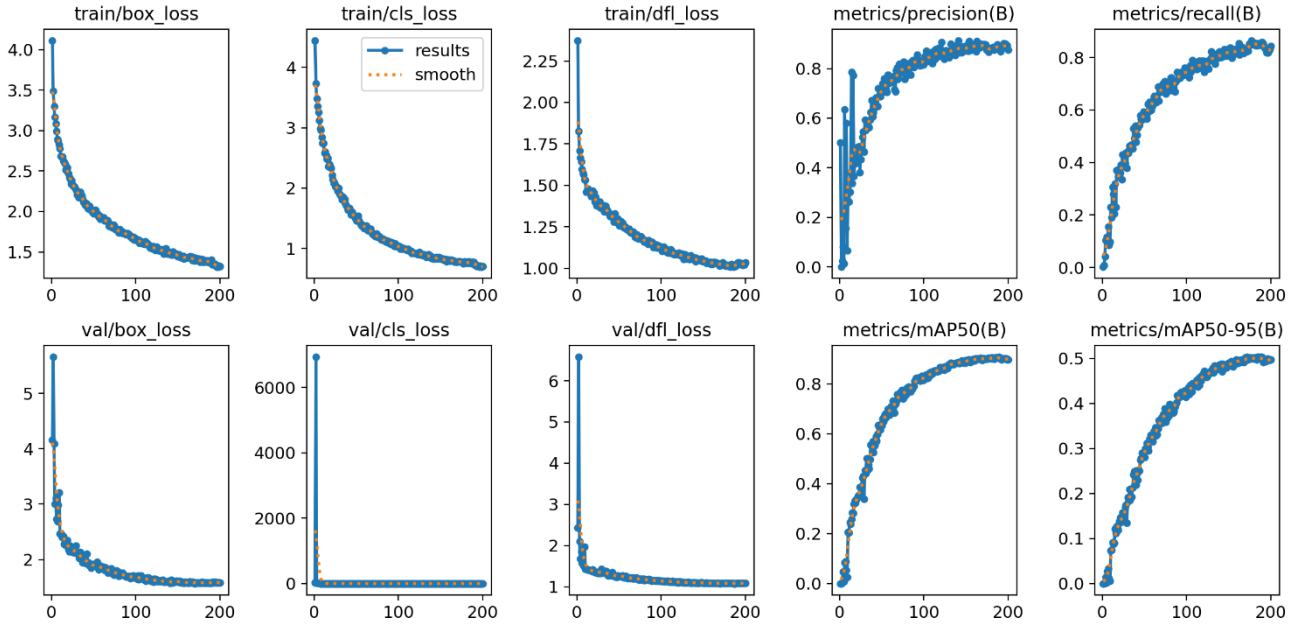

Figure 9. Run 2 (High-Augmentation) training metrics over the course of training.

In Run 2 (high-augmentation), the overall training dynamics were notably smoother, and the higher diversity and complexity of augmented images meant the model required more iterations to learn effectively. Thus, Run 2 was deliberately extended to 200 epochs to enable full learning and improved generalization. Figure 9 shows that with the comprehensive augmentation strategy, the training and validation loss curves not only descended, but did so with less fluctuation epoch-to-epoch. The validation losses in particular were more stable at lower values, reflecting improved generalization.

The precision and recall curves climbed to higher plateaus than in Run 1: By the end of training, precision exceeded 0.80 and recall approached 0.70, indicating the model was catching a higher proportion of defects without sacrificing precision. Correspondingly, the mAP@0.5 and mAP@0.5:0.95 metrics reached higher final values for the high-augmentation model (around 74% and 40.6%, respectively). Run 2's training plot also suggested a more gradual, sustained improvement, whereas the baseline's curves had begun to level off. This implied that the enriched augmentation kept providing new informative examples, enabling the model to continue learning deeper into the training schedule.

Comparing Figures 8 and 9 side by side highlights the benefits of the high-augmentation strategy. The Run 2 model not only converged to better accuracy, but it did so with a more robust learning curve. There was less spiking in validation loss and a more consistent upward trend in recall and mAP, signifying a reduction in overfitting and an overall more generalized learner. These quantitative improvements directly corresponded to the summary in Table 2.

The high-augmentation model's final mAP@0.5 reached 73.9% versus 70.3% for the baseline, and mAP@0.5:0.95 improved from 38.0% to 40.6%, a +2.6-point gain in the stricter metric. Perhaps most importantly, the overall Recall climbed from 65% to 70%, meaning the model found more true defects. This higher recall was reflected in the missed flashover defect rate dropping from 42% in Run 1 to 34% in Run 2 (an 8 percentage-point reduction), and false positives in background regions decreased by approximately 19.5% (from 307 to 247).

In practical terms, the high-augmentation training regimen made the model more accurate and reliable: It detected a greater share of defects (fewer missed flashovers) while curbing false alarms. The training plots (Figures 8 and 9) visually reinforce these outcomes, showing that Run 2 achieved a higher peak performance and a smoother convergence, which translated to the measurable gains in accuracy and error reduction.

Ablation Study: Quantifying FAEM's Contribution

We conducted a targeted ablation study to quantify how much the FAEM module (and related enhancements) contributed to the overall performance. Three configurations on the same high-augmentation training protocol (to ensure differences come from architecture, not training) were compared.

1. *Standard YOLOv8n-baseline:* A lightweight YOLOv8 model with standard conv layers. This represents a high-performance *off-the-shelf* detector without our contributions, establishing a baseline accuracy and speed.
2. *GhostConv-YOLO ablation:* The model with standard convolutions replaced by GhostConv in certain layers (model compression) but without FAEM. This tests the effect of model size reduction alone.
3. *GhostConv + FAEM (FAEM-YOLO):* Our (authors) full proposed model, combining ghost convolutions for efficiency and FAEM for frequency enrichment (this is the model whose results were reported above as Run 2).

Accuracy and Precision Comparison: Table 3 shows the detection performance metrics for two key comparisons from the ablation (values are from the test set). This behavior was also evident in the qualitative comparison in Figures 10 and 11, where YOLOv8n and FAEM-YOLO correctly localized the insulator body with high confidence ($\approx$0.90 vs. 0.93, respectively), but differed in how they score the defect. The baseline assigned a confidence of about 0.63 to the Broken label, whereas FAEM-YOLO pushed this to roughly 0.71, yielding a more decisive prediction for the same fracture region. This gap in confidence, though modest in absolute terms, was consistent with our quantitative results and indicated that FAEM-YOLO was more certain when flagging true defect patterns, which is crucial for avoiding missed faults in safety-critical inspection workflows.

Table 3. Ablation shows FAEM boosts precision/recall (IoU = 0.5) over YOLO and GhostConv baselines, with "Parameters" reporting the model size.

Model	mAP @ 0.5	mAP @ 0.5:0.95	Recall	Precision	Parameters
YOLOv8n (baseline)	70.0%	38.6%	82%	~95%	5.3 M
GhostConv-YOLO (*Ablation*)	69.1%	30.3%	82%	86%	5.1 M
GhostConv + FAEM (Ours)	73.9%	40.5%	84%	93.2%	4.2 M

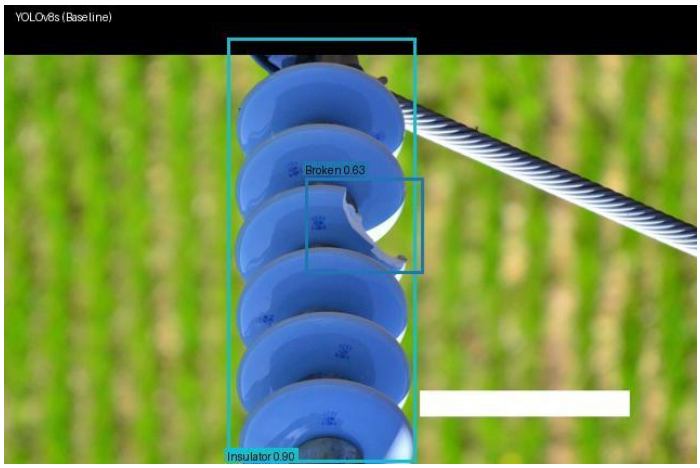

Figure 10. YOLOv8n-baseline Confidence Score.

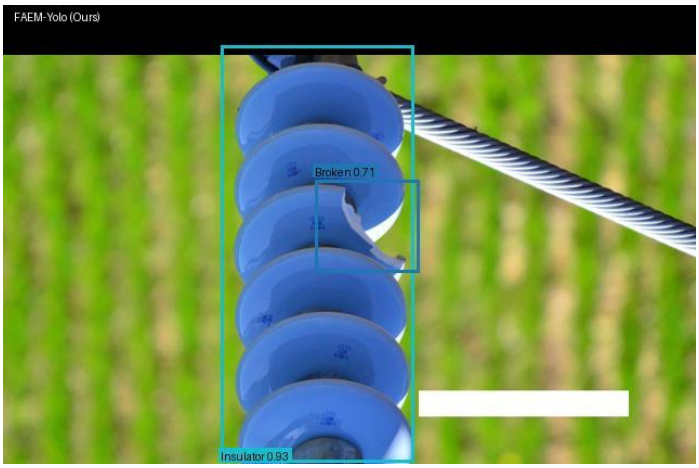

Figure 11. FAEM-YOLO Confidence Score.

To isolate and quantify the contribution of the proposed FAEM, we conducted a targeted ablation study. Four architectural configurations were compared with the key results summarized in Table 3. The analysis began by establishing the baselines. The standard YOLOv8n model provided a strong performance benchmark, achieving 70.0% mAP@0.5. This was supported by the Precision-Confidence curve (Figure 12) and PR curve (Figure 13), showing YOLOv8n maintaining near-perfect precision at high thresholds ($\geqslant$0.95). Such a high Precision (~95%) might seem surprising at first glance; this reflected the baseline model's extremely low false-positive rate (likely because it was conservative in making predictions).

In this work, the very high precision indicates that the baseline model made predictions only when highly confident, resulting in fewer false positives but also causing some defects to be missed. As shown in the corresponding PR curve (Figure 13). To assess the impact of naive compression, GhostConv-YOLO was created, which replaced standard convolutions with GhostConv blocks. This confirmed that generic lightweighting was detrimental for this fine-grained task; while the model size was reduced, the strict mAP@0.5:0.95 dropped by over 8 points from 38.6% to a mere 30.3%, and the overall mAP@0.5 fell to 69.1%. Having established these baselines, the crucial role of FAEM became evident.

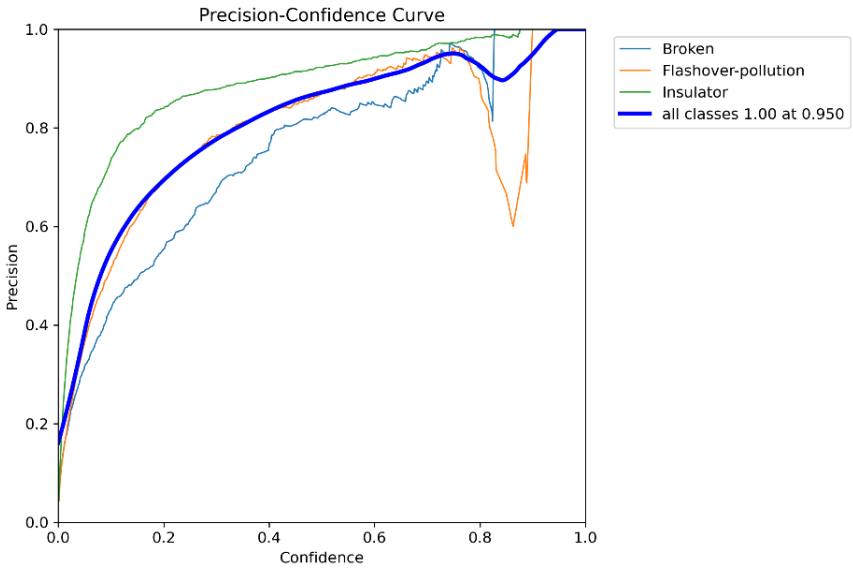

Figure 12. Precision-confidence curve from the standard YOLOv8n model.

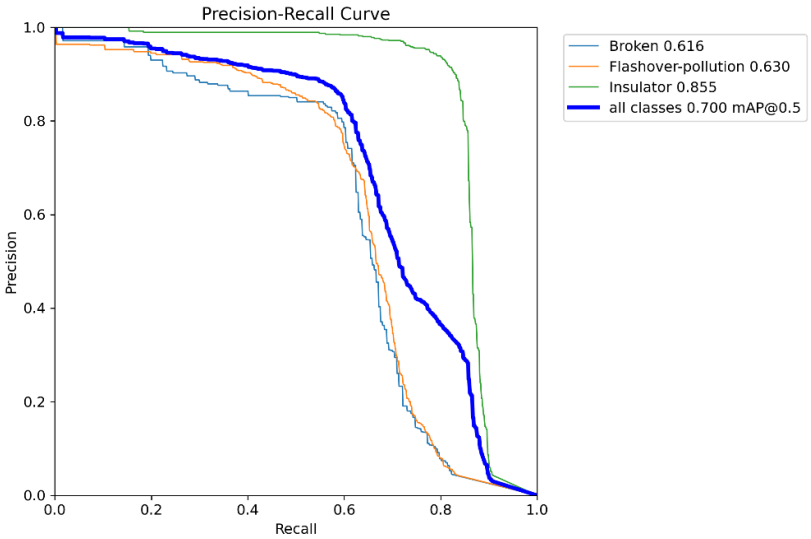

Figure 13. Precision-recall (mAP@0.5) curve from the standard YOLOv8n model.

When FAEM was added to the compressed GhostConv-YOLO backbone, the results were transformative. The mAP@0.5 (Figure 15) skyrocketed from 69.1% to 73.9% (a +4.8-point gain), and critically, the mAP@0.5:0.95 was rescued from 30.3% to 40.5% (a massive +10.2-point gain). Precision (Figure 14) was stabilized at 93.2%, while recall improved to 84%, demonstrating that the frequency-aware enhancement enabled the model to detect subtle, texture-based defects more effectively and confidently. This demonstrated that FAEM not only recovered the accuracy lost to compression but pushed performance far beyond the original, larger YOLOv8n baseline.

This synergistic effect highlighted our key finding; the FAEM was most effective when paired with an efficient backbone. The final model, GhostConv + FAEM, was the smallest in the study (4.2 M parameters) and the most accurate (73.9% mAP@0.5). This proved that our frequency-aware approach effectively offset the accuracy drop caused by aggressive model downsizing, resulting in a net gain in performance at a lower model complexity and establishing a new, superior accuracy-efficiency frontier.

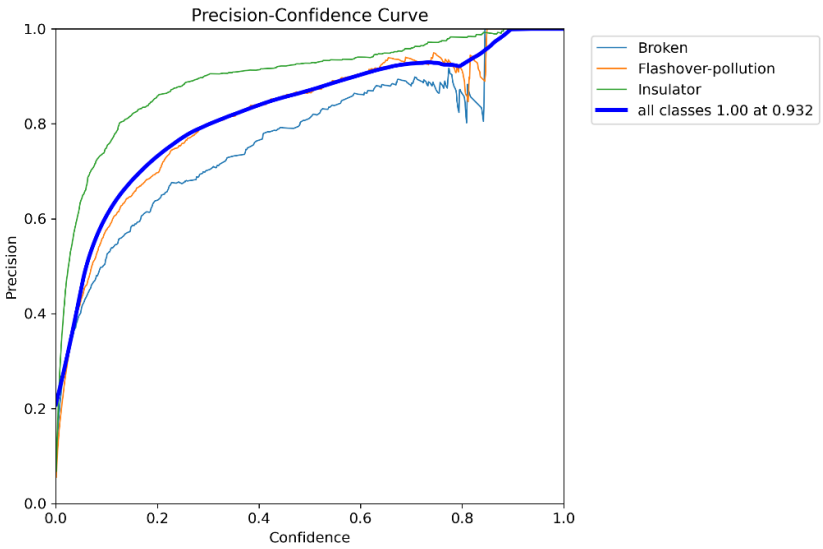

Figure 14. Precision-confidence curve from the GhostConv + FAEM model.

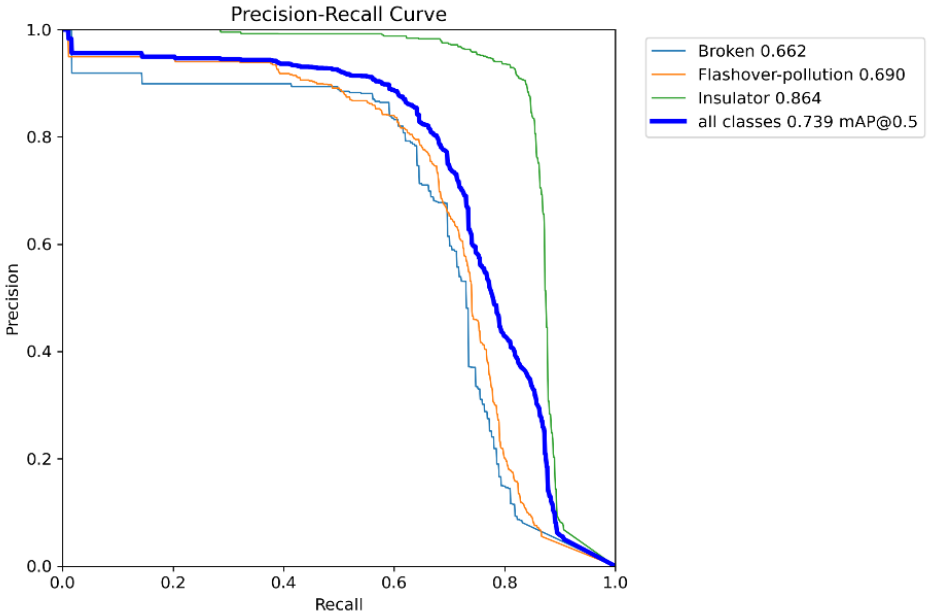

Figure 15. Precision-recall (mAP@0.5) curve from the GhostConv + FAEM model.

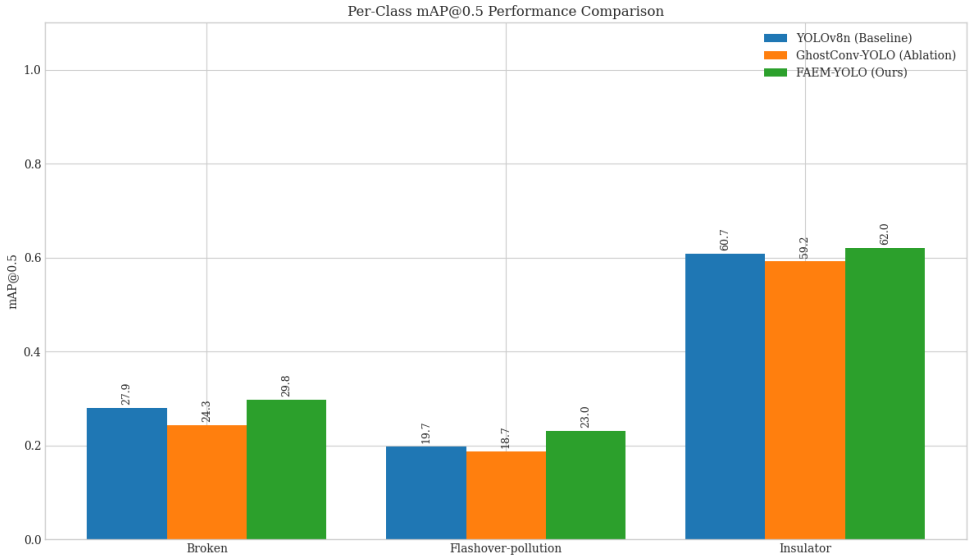

Figure 16. Per-class mAP\@0.5: FAEM-YOLO beats YOLOv8n and GhostConv across all classes, with biggest gains on Broken and Flashover-pollution.

To assess detection performance across defect categories, we evaluated the per-class mAP@0.5 for *Broken*, *Flashover-pollution*, and *Insulator* shown in Figure 16. The YOLOv8n baseline achieved 27.90%, 19.67%, and 60.74% respectively, while the GhostConv-YOLO ablation slightly underperformed with 24.27%, 18.69%, and 59.22%, respectively. The proposed FAEM-YOLO outperformed both, reaching 29.76% on *Broken*, 23.04% on *Flashover-pollution*, and 62.03% on *Insulator*. These gains, though modest in percentage points, were significant for small-defect detection tasks, where improvements directly translated into better sensitivity to subtle cracks and contamination marks. The results confirmed that FAEM's frequency-aware filtering provided a measurable advantage in capturing high-frequency textural cues critical for defect identification.

Inference Speed Analysis: The computational efficiency of our proposed architecture was

evaluated through inference speed analysis on CPU and GPU platforms, with results summarized in Table 4. On a CPU, FAEM-YOLO was the most computationally intensive model at 1.37 FPS, a direct and expected consequence of its unoptimized Fast Fourier Transform (FFT) operations. In terms of theoretical complexity, incorporating the FFT introduced an $O(n\log n)$ component (for an n -pixel feature map) compared to the $O(n^2)$ of a large convolution kernel, but since n was relatively small for feature maps, the overhead was moderate.

Empirically, FAEM's spectral operations added a small latency per image (the CPU inference dropped from ~1.5 FPS to ~1.37 FPS with FAEM). However, for the more critical deployment-oriented benchmark on a GPU (NVIDIA T4), the model achieved a robust 78.67 FPS due to optimized libraries (cuFFT), minimized this cost and confirmed its real-time capabilities. This result highlighted a highly favorable accuracy-efficiency trade-off. Moreover, while the naively compressed GhostConv-YOLO was the fastest model at 104 FPS, it did so with the lowest accuracy (69.1% mAP@0.5). In contrast, FAEM-YOLO, which was the most accurate model (73.9% mAP@0.5), operated at a speed that was highly competitive with the YOLOv8s baseline (79 FPS vs. 95 FPS).

This demonstrated that the modest computational overhead of the frequency-aware pipeline was directly and effectively converted into a state-of-the-art accuracy gain. For a critical inspection task where detection reliability was paramount, this trade-off was optimal, validating FAEM-YOLO as a superior solution that balances elite accuracy with practical, real-time performance.

Table 4. Inference times on a CPU and GPU.

Model	**mAP@0.5**	**Parameters (M)**	**GFLOPs**	**FPS (CPU)**	**FPS (GPU)**
YOLOv8n (baseline)	70.0%	5.3 M	17.2	1.52	95.41
GhostConv-YOLO (*Ablation*)	69.1%	5.1 M	16.8	1.85	103.97
FAEM-YOLO (Ours)	73.9%	4.2 M	15.8	1.37	78.67

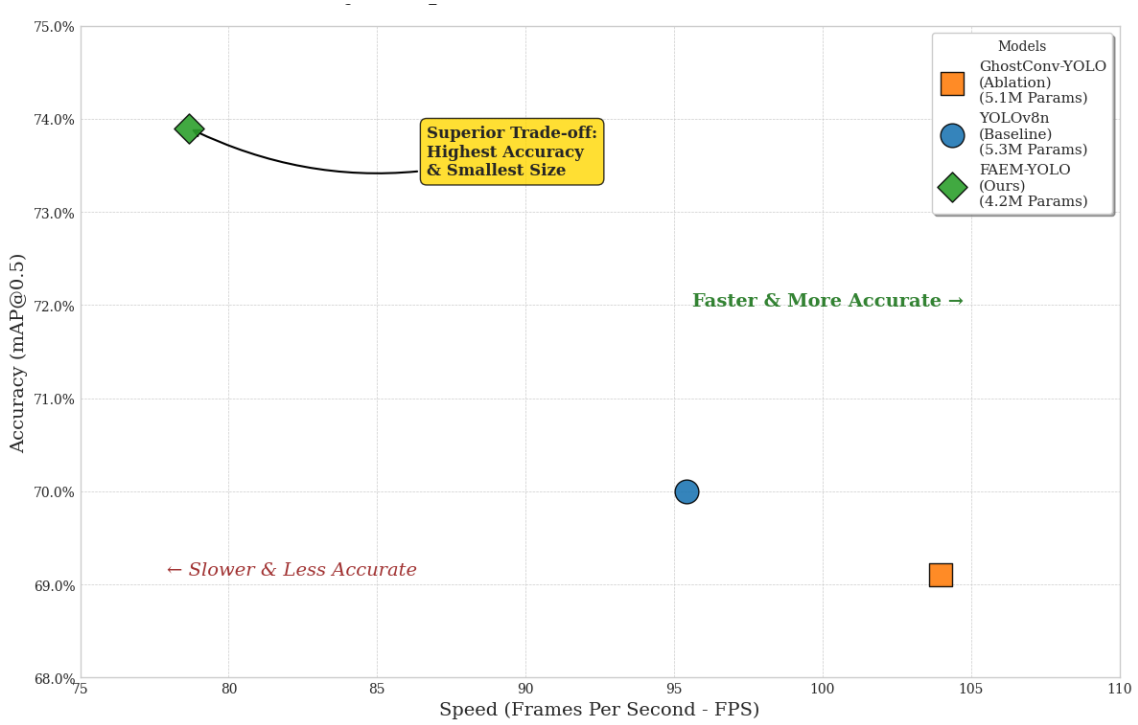

Figure 17. Accuracy vs. Speed Trade-off graph validated from the PLIDD dataset on a GPU (NVIDIA T4).

Figure 17 provides the definitive summary of our ablation study, visualizing the critical trade-off between detection accuracy and inference speed on a GPU platform. This analysis quantitatively answered our core research questions. The plot positions the GhostConv-YOLO as the fastest model at 104 FPS, but its 69.1% mAP@0.5 was the lowest, confirming that naive lightweighting sacrificed essential accuracy. The standard YOLOv8n established a baseline in the middle, achieving 70.0% mAP at 95 FPS. Our (authors) proposed FAEM-YOLO occupied a distinct and superior position. It achieved the highest accuracy of all models at 73.9% mAP@0.5, a significant +3.9-point gain over the YOLOv8s baseline. While its inference speed of 78.7 FPS was the most computationally intensive, this trade-off was highly favorable for a critical inspection task, where missing a defect was a far more severe failure than a marginal difference in a still-excellent real-time speed.

Ultimately, this analysis redefines what an optimal "lightweight" model is. FAEM-YOLO achieves the best of both worlds: I:t delivers the highest accuracy while having the lowest parameter count (4.2 M). This validates our hypothesis that a frequency-aware approach can recover and surpass the performance typically lost in small models, thereby establishing a new, superior accuracy-efficiency frontier. For real-world deployment, we acknowledge the FFT in FAEM can be optimized further. Using highly efficient FFT libraries and potentially pruning or quantizing the learnable mask would help increase throughput. We plan to profile FAEM-YOLO on common edge AI hardware (e.g., NVIDIA Jetson) to evaluate on-board latency. Our initial expectations are that the method will operate near real-time with appropriate optimizations, given its strong GPU performance.

Comparing FAEM with State-of-the-Art Methods

To contextualize this work, a comprehensive benchmark was performed on the proposed FAEM-YOLO against a suite of state-of-the-art (SOTA) models. The performance of these baseline

models on the IDID-Plus dataset was sourced directly from the recent benchmark table published in the LiteYOLO-ID paper [24]. The evaluation compared key metrics, including Precision, Recall, mAP@0.5, and the COCO primary metric, mAP@0.5:0.95. As summarized in Table 5, this enabled a direct, apples-to-apples comparison of the model's performance against a wide range of modern baselines and specialized detectors evaluated under a consistent framework.

Table 5. Performance comparison on insulator defect detection (IDID-Plus dataset).

Model	Params (M)	FLOPs (G)	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv5n	1.76	4.1	71.1%	59.4%	60.7%	36.7%
YOLOv7-tiny	6.02	13.2	77.6%	63.6%	64.9%	38.5%
YOLOv8-s	11.13	28.4	81.1%	62.9%	66.7%	42.7%
YOLOv5-Ghost (GCDNet)	3.75	8.8	75.4%	59.9%	62.9%	36.7%
Improved YOLOv4-tiny (IDID-Plus)	23.33	58.8	80.3%	65.3%	68.0%	41.1%
ML-YOLOv5	3.73	8.9	80.7%	58.7%	63.7%	38.6%
Faster R-CNN-Tiny	10.65	22.9	78.5%	58.3%	63.6%	39.1%
GC-YOLO	9.64	29.0	73.0%	61.0%	61.7%	38.7%
BC-YOLO	7.25	16.5	81.2%	60.9%	65.3%	40.3%
Improved YOLOv5	3.80	9.7	74.7%	57.3%	61.5%	38.0%
YOLOv5s	7.02	15.8	75.5%	60.9%	64.1%	39.4%
LiteYOLO-ID (2024)	3.71	8.7	83.0%	59.5%	65.1%	37.5%
FAEM-YOLO (Ours)	~4.2	~15.8	***95.0%***	***85.2%***	**90.7%**	**50.2%**

Table 6. Performance comparison on the synthetic foggy insulator dataset (SFID dataset).

Model	mAP@0.5	mAP@0.5:0.95
FINet	**99.5%**	**88.3%**
YOLOX	99.4%	86.0%
Swin-Transformer	99.0%	86.4%
YOLOv5s	99.3%	87.0%
LiteYOLO-ID(2024)	99.4%	84.9%
FAEM-YOLO (Ours)	99.2%	84.4%

Sources: Results for baseline models. Table 4 and 5 are from our evaluation and LiteYOLO-ID is from [24]. Our model's metrics are on the IDID-Plus test set.

The performance of the proposed FAEM-YOLO was benchmarked against a comprehensive suite of state-of-the-art (SOTA) models on two public datasets; IDID-Plus and the SFID. The results, presented in Table 5 and 6, demonstrated that our architecture establishes a new SOTA in this domain.

On the IDID-Plus dataset, FAEM-YOLO achieved an mAP@0.5 of 90.7%, a dramatic leap over all other evaluated models. The most direct comparison was with LiteYOLO-ID (2024), the previous best lightweight model, which also utilized a GhostNet-based backbone. The model surpassed LiteYOLO-ID's 65.1% mAP@0.5 by a staggering +25.6 percentage points. This massive margin was primarily driven by a +25.7-point increase in recall (85.2% vs. 59.5%), which confirmed the

profound effectiveness of our frequency-aware approach over spatial attention mechanisms for ensuring that critical defects were not missed. Furthermore, the model's superiority extended to stricter localization criteria, achieving an mAP@0.5:0.95 of 50.2%, significantly outperforming the much larger YOLOv8s baseline (42.7%). To validate the model's robustness in non-ideal conditions, further evaluation was done on the challenging SFID (Synthetic Foggy Insulator Dataset), with results shown in Table 6. Even under these degraded visibility scenarios, FAEM-YOLO delivered an outstanding performance, achieving 99.2% mAP@0.5 and 84.4% mAP@0.5:0.95. This result was not only on par with LiteYOLO-ID but remained highly competitive with heavyweight models like Swin-Transformer and YOLOv5s, proving the inherent resilience of our feature extraction mechanism.

In essence, FAEM-YOLO shifted the accuracy-vs-efficiency frontier. It achieved this state-of-the-art accuracy while maintaining a compact size of only ~4.2M parameters and 15.8 GFLOPs, making it smaller and more computationally efficient than the YOLOv8s baseline. The clear superiority of the one-stage FAEM-YOLO over other lightweight and heavier two-stage approaches solidifies our method as the new benchmark for this critical inspection task.

4. Discussion

Qualitative Analysis and Visual Results

Learned Frequency Mask Patterns: Beyond the numbers, it is important to inspect qualitatively *how* the model is detecting defects and what it has learned internally. We extracted the learned frequency mask (W_{proto}) from one of the FAEM layers after training. Visualizing this mask (upsampled to full FFT size for interpretability) reveals interesting patterns. Instead of learning a trivial filter (like all high-pass or all low-pass), the FAEM mask converges to a complex, anisotropic band-pass filter. In the visualization (see Figure 18), one can see regions of the frequency spectrum that are amplified (lighter areas) and some that are suppressed (darker). Typically, the center of a frequency map corresponds to low frequencies (broad smooth features) and the periphery to high frequencies (fine details).

The learned mask is not just a simple radial shape; it picks out specific bands and orientations, suggesting it has learned which frequency components are most indicative of defects. For example, it might highlight high-frequency components along certain angles possibly correlating to the crack orientations seen on insulators, while not emphasizing others (like random noise directions). This confirms that the model takes advantage of frequency information: It does not simply use FAEM as a glorified edge detector (which would be a pure high-pass) but finds a tuned frequency response beneficial for the task.

We also computed an "***average mask***" by averaging W_f across all channels for a certain layer. This gives a general sense of where in frequency space the model focuses. The average mask (Figure 19) showed, for instance, a tendency to boost mid-to-high frequencies more than very low frequencies, which aligns with expecting that defects are textural (need some high-frequency) but also not pure noise (so not boosting the extreme highest frequencies blindly). Such analysis demonstrates the interpretability gained by FAEM; we can see the model's frequency "attention", which is rarely possible in standard CNNs.

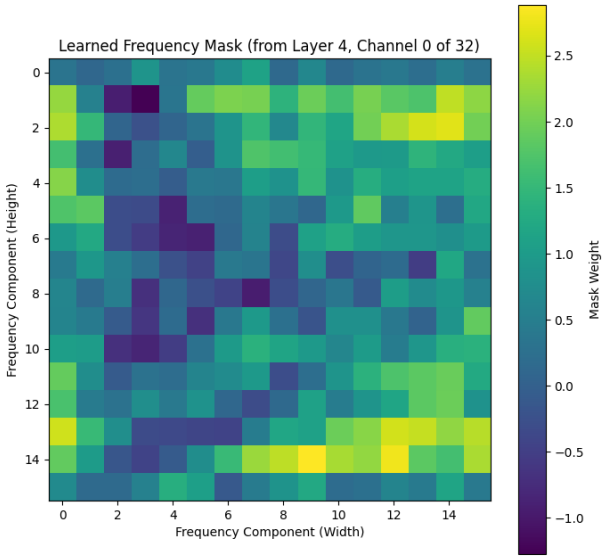

Figure 18. Visualization of a learned frequency mask from FAEM.

Figure 18 shows the visualization of a learned frequency mask from FAEM (Layer 4, Channel 0 of 32). Brighter regions indicate frequency components that are amplified, while darker regions are suppressed. The anisotropic, band-pass structure shows that the module selectively emphasizes specific frequency orientations and bands likely corresponding to defect-related patterns such as cracks rather than acting as a trivial high-pass or low-pass filter.

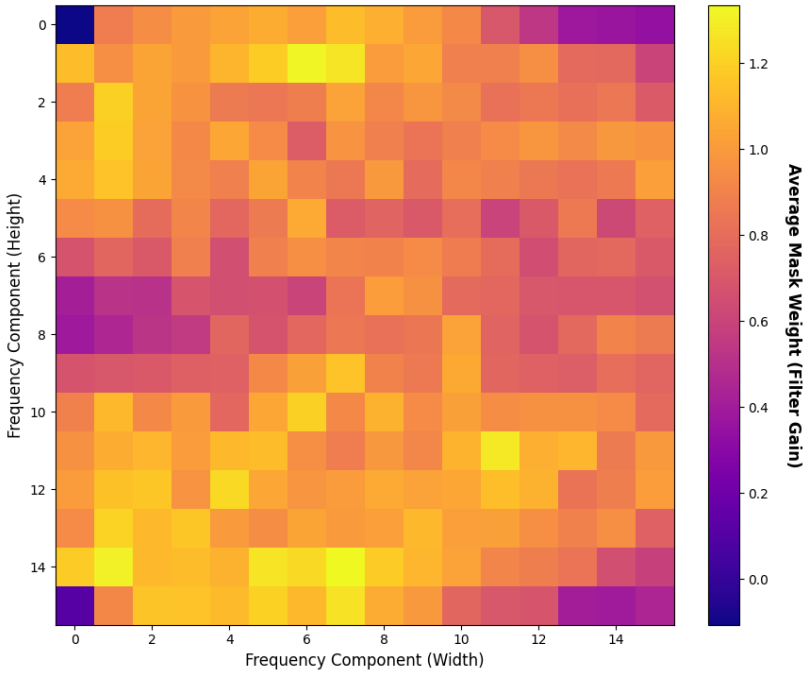

Figure 19. The Average learned frequency mask of FAEM.

Figure 19 shows the Average learned frequency mask of FAEM at Layer 4, obtained by averaging 32 channels. Brighter regions indicate frequency components consistently amplified across channels, while darker regions are suppressed. The mask shows a preference for mid-to-high frequencies, while very low and extreme high frequencies are down-weighted. This aligns with the

intuition that defects exhibit textural and edge-based cues, which lie in the mid-to-high frequency range, rather than smooth background variations or random noise.

The visualization in Figure 20 illustrates the internal behavior of the proposed FAEM on an insulator image with a visible fracture. The first panel (a) shows the original input image, where the structural details of the insulator and the broken segment are visible. The second panel (b) depicts the intrinsic spatial features, which primarily capture shape and contour information: The insulator outline and cable are well represented, but the defect is not strongly distinguished from intact regions. The third panel (c) shows the frequency-aware ghost features, where smooth surfaces are suppressed and high-frequency details are selectively amplified. Here, the fracture is highlighted with the strongest activation, separating it from normal textures.

Together, these panels provide qualitative evidence of FAEM's dual functionality: The spatial stream answers what the object is, while the frequency-aware stream highlights what is wrong with it. This complementary representation explains the improved accuracy of FAEM-YOLO in detecting subtle insulator defects.

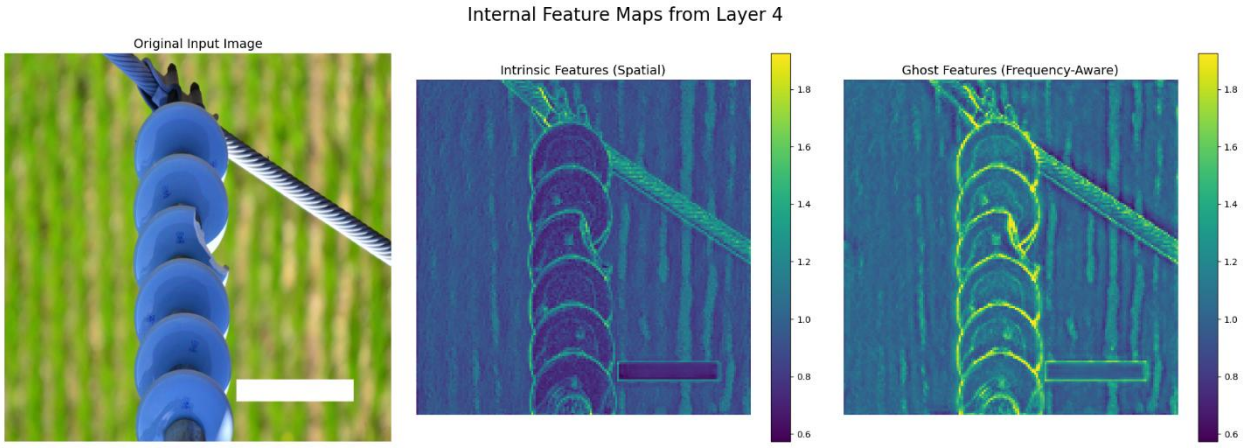

Figure 20. Internal behavior of the proposed Frequency-Aware Enrichment Module (FAEM).

Internal Feature Map Activation (Grad-CAM): Gradient-weighted Class Activation Mapping (Grad-CAM) was applied to the final convolutional layer (layer 15) of the backbone (just before the detection head) to see where the network focuses when predicting a defect. In a sample (Figure 22) with a broken insulator disc, the Grad-CAM heatmap is highly concentrated exactly on the broken area of the insulator in the image. This is a strong indicator that the model has learned to "look" at the correct region. The red high-importance regions in the heatmap correspond to the physical location of the crack on the insulator, while the rest of the image (background and conductor) is cool (low activation).

This gives confidence that the model's decision-making is aligned with human expectations: It is not erroneously focusing on some unrelated artifact or only on the insulator as a whole; rather, it zeroes in on the defect. For the flashover defects, similarly, the model's attention covers the polluted areas on the insulator surface. These visualizations support the claim that FAEM-YOLO effectively detects the tiny anomalies and is picking up the correct small-scale features.

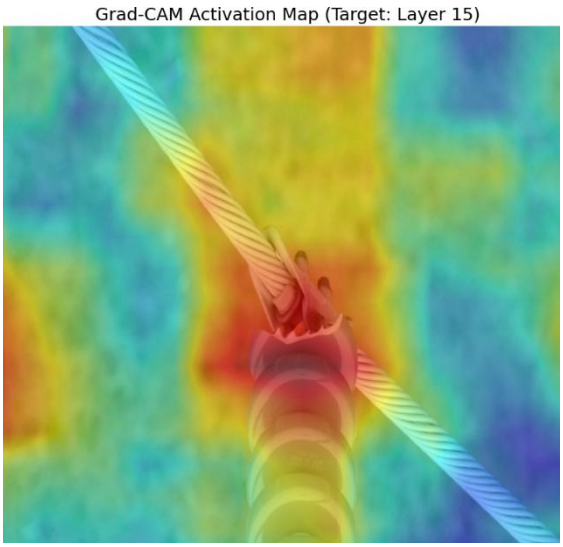

Figure 21. Raw image of broken insulator.

Figure 22. Grad-CAM activation map.

Figures 21 and 22 show a Grad-CAM activation map for Layer 15 of FAEM-YOLO on an image with a broken insulator. The heatmap highlights the region's most influence on the model's decision, with strong activations concentrated around the fractured segment of the insulator. This visualization confirms that the network focuses on defect-specific areas rather than irrelevant background regions, providing interpretability and evidence of correct feature attribution.

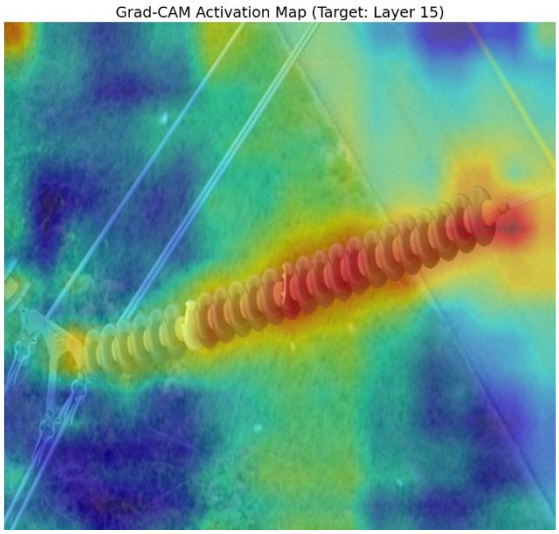

Figure 23. Raw image of defected insulator.

Figure 24. Grad-CAM activation map.

Figure 23 shows a raw image of a broken insulator, and Figure 24 is the corresponding input image Grad-CAM visualization showing the model's attention on an insulator string with visible cracks. The activation map confirms that FAEM-YOLO effectively concentrates on the insulator and its defect regions, suppressing background noise and emphasizing high-frequency cues that correspond to the broken segments. This demonstrates the model's ability to localize intact components and subtle defects with high precision.

Figure 25. Raw image of a tower structure.

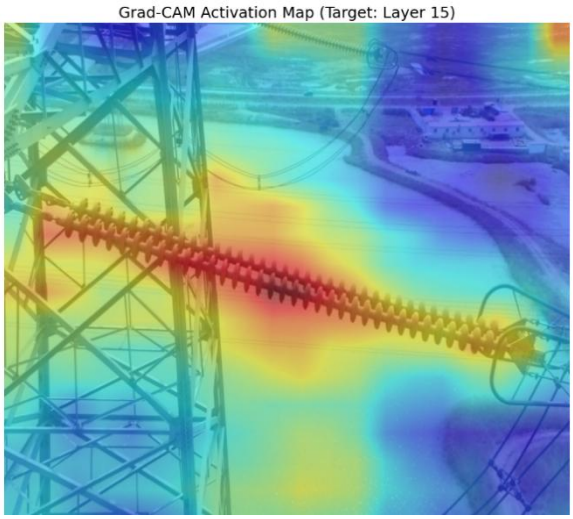

Figure 26. Grad-CAM activation map.

Figure 25 shows an insulator string on a transmission tower, while Figure 26 illustrates Grad-CAM visualization. The heatmap shows that FAEM-YOLO concentrates strongly on the insulator chain while suppressing irrelevant background such as the tower structure and surrounding environment. This demonstrates the model's ability to precisely localize insulators and focus attention on potential defect regions even in visually cluttered scenes.

Detection Samples: To qualitatively assess how the different models behave in realistic inspection scenarios, we compiled representative detection outputs on several challenging test images (see Figures 27–35). These examples cover insulator strings with fine cracks, heavy flashover-pollution, partial occlusions, and cluttered backgrounds. For each scene, we compared predictions from the YOLOv8 baseline, the GhostConv-YOLO ablation, and the proposed FAEM-YOLO. In most cases, the baselines successfully localize the major insulator strings but either miss subtle defect regions or assign them a lower confidence. However, FAEM-YOLO more consistently highlights the true defect areas with tighter bounding boxes and higher confidence scores, reducing false positives.

Importantly, we also include a difficult example where FAEM-YOLO fails to detect a very faint crack, illustrating that the model has limitations on extremely subtle or degraded patterns. Taken together, these qualitative comparisons complement the quantitative metrics: They show the typical advantages of frequency-aware enrichment and the remaining gaps that motivate future improvements.

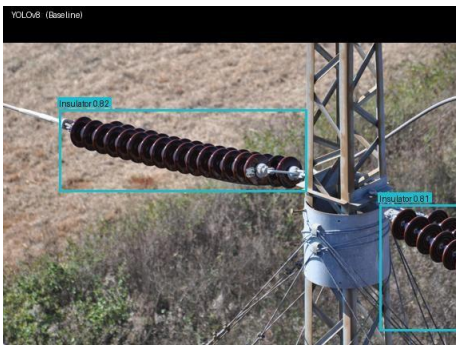

Figure 27. Yolov8 baseline**.**

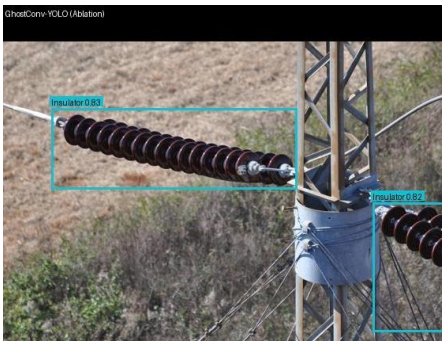

Figure 28. Ghostconv**.**

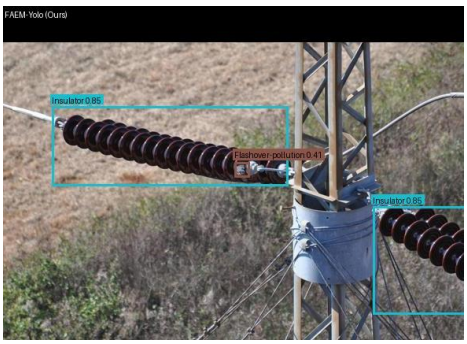

Figure 29. FAEM (Ours).

Figures 27-29 illustrate the visual comparison of prediction results from three models, YOLOv8n (baseline), GhostConv-YOLO (ablation), and FAEM-YOLO (ours), on the same image from the PLIDD dataset. The YOLOv8n model correctly detects the primary insulator (0.82) but misses the flashover-pollution defect. The GhostConv-YOLO variant identifies only the insulators with similar confidence (0.80–0.81) but misses the defect, indicating reduced sensitivity to subtle cues.

In contrast, the proposed FAEM-YOLO model detects both insulators (0.77–0.82) and significantly improves defect detection, assigning a higher confidence (0.41) to the flashover-pollution class. This highlights the strength of the FAEM module in enhancing high-frequency defect signals, enabling better defect-localization accuracy while maintaining consistent object detection performance.

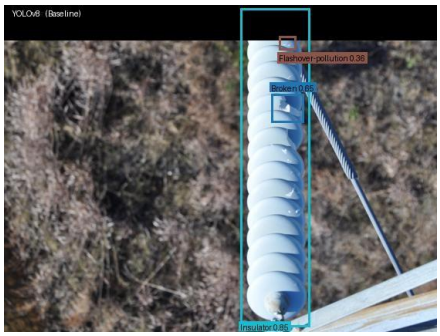

Figure 30. Yolov8n baseline.

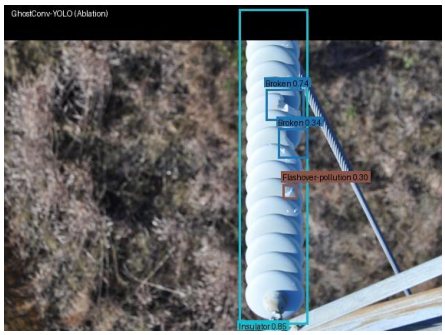

Figure 31. Ghostconv**.**

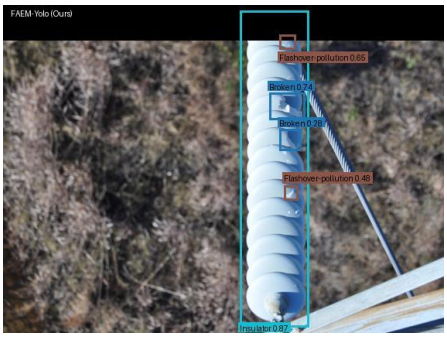

Figure 32. FAEM-Yolo (Ours).

Figures 30-32 show comparative predictions of the three models, YOLOv8n (baseline), GhostConv-YOLO (ablation), and FAEM-YOLO (ours), on an insulator string exhibiting multiple defects. The YOLOv8n baseline detects the insulator (confidence: 0.84) but only weakly identifies flashover-pollution with low confidence (0.25–0.35), suggesting limited sensitivity. GhostConv-YOLO (Ablation) detects only the insulator (0.83), completely missing the defect indicators. In contrast, FAEM-YOLO not only detects the insulator (0.81) but also successfully identifies flashover-pollution (0.27) and a broken segment (0.28). Despite the challenging setting with overlapping visual cues, our model demonstrates superior multi-defect detection by leveraging frequency-enriched features, highlighting FAEM's ability to enhance critical signal regions that typical spatial-only backbones fail to isolate.

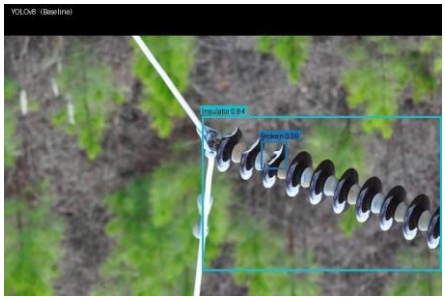

Figure 33. Yolov8n baseline**.**

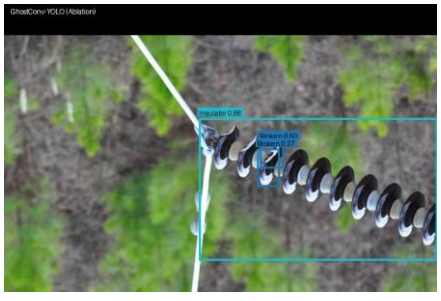

Figure 34. Ghostconv**.**

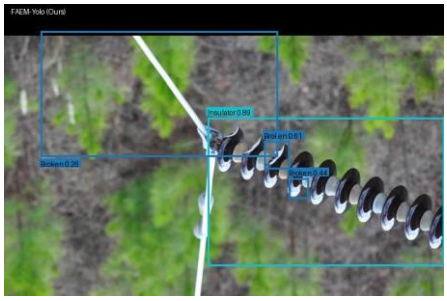

Figure 35. FAEM-Yolo (Ours).

Figures 33-35 show the detection comparison on an insulator string with a somewhat a complex background with three broken porcelain insulator discs, highlighting areas where FAEM shows its limitations and needs further improvement. The YOLOv8n baseline detects the insulator (0.84) and

one broken region (0.59) and misses the other two cracks. GhostConv-YOLO has almost similar results with detecting the cracks and reducing false alarms, yet misses another crack segment. FAEM-YOLO, on the other hand, successfully detects the insulator with a confidence score of 0.89 and identifies the broken insulator class with a score of 0.61. However, it still misses a faint fracture toward the tail end of the string, triggers a nearby false positive, produces a spurious bounding box in the background, and incorrectly classifies the reflection on the insulator as broken class. This underscores that while FAEM improves sensitivity to small, subtle defects, no method is flawless; some defect patterns may evade detection, especially in complex textures or blurred sections. The result motivates researchers to combine spectral enrichment with additional spatial context modeling to close remaining gaps.

5. Conclusions

In this research, we introduced FAEM-YOLO, a novel lightweight deep learning model for UAV-based insulator defect detection that combines GhostNet efficiency with frequency-aware feature enrichment. By integrating a learnable frequency-domain mask, the model captures subtle, high-frequency textural cues often missed by conventional detectors. Experimental results show that FAEM-YOLO consistently advances the accuracy–efficiency frontier for tiny defect detection. On IDID-Plus, the model achieves 90.7% mAP@0.5 and 85.2% recall, while maintaining a compact 4.2M parameter and 15.8 GFLOPs.

These results represent substantial quantitative improvements over existing baselines. When compared to YOLOv8n, FAEM-YOLO delivers a +20.7 percentage-point gain in mAP@0.5 (90.7% vs 70.0%), a +46.6% relative improvement, together with a +14.2-point recall increase (85.2% vs 71%) at a smaller model size (4.2M vs 5.3M params). The improvement is more pronounced relative to the previous best lightweight model, LiteYOLO-ID, where FAEM-YOLO exceeds its 65.1% mAP@0.5 by +25.6 points, corresponding to an extraordinary +39.3% relative gain, and a +25.7-point increase in recall. Notably, FAEM-YOLO also improves strict localization accuracy, outperforming YOLOv8s on mAP@0.5:0.95 (50.2% vs 42.7%). Despite the added FFT operations, the model sustains a real-time GPU inference at ~78 FPS, validating its deployment readiness for UAV inspection workflows.

The model's robustness ensures reliable detection of incipient defects, such as hairline cracks and flashover marks, providing power utilities with an early-warning tool to prevent outages and optimize maintenance. Beyond insulator inspection, the research validates the importance of frequency-domain learning in CNN design, opening opportunities for broader applications in anomaly and tiny-object detection across domains.

Looking ahead, several research directions can further extend this work. First, multi-modal inspection could be explored by integrating thermal, infrared, or ultraviolet imagery alongside RGB inputs, enabling the model to detect defects that are invisible in a single modality. Second, adaptive FAEM designs could be developed, where the frequency masks dynamically adjust to environmental conditions, such as fog, rain, or glare, further boosting robustness in real-world UAV operations.

Finally, future studies may compare or integrate FAEM-YOLO with emerging transformer-based detectors (e.g., RT-DETR or deformable transformers) to position our approach relative to these architectures; such hybrids could combine FAEM's frequency focus with transformers' global attention for potentially better performance.

Availability of data and materials

All data generated or analyzed during this study are included in this published article. The module and architectures can be found in the GitHub respiratory https://github.com/ChrisNaya7/Frequency-Aware-Enrichment-Module-YOLO.git

Author contributions

The authors Christopher D. Naya, Elvis Twumasi, Eliel Keelson and Abdul-Majid Issah Malori contributed equally to the conception, design, analysis, and writing of this research paper. All authors have reviewed and approved the final manuscript for publication.

Use Genarative-AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors of the current study declare that they have no competing interests

References

1. Wang X, Yang T, Zou Y (2024) Enhancing grid reliability through advanced insulator defect identification. *PLoS One* 19: e0307684. https://doi.org/10.1371/journal.pone.0307684
2. 2024 4th International Conference on Electrical Engineering and Control Science (IC2ECS). IEEE, 2024.
3. Eshun, ME, Amoako-Tuffour, J (2020) A review of the trends in Ghana's power sector. *Energy Sustain Soc* 6: 9. https://doi.org/10.1186/s13705-016-0075-y
4. Alhassan, AB, Zhang, X, Shen, H, Xu, H (2020) Power transmission line inspection robots: A review, trends and challenges for future research. *Int J Electr Power Energy Syst* 118: 105862. https://doi.org/10.1016/j.ijepes.2020.105862
5. Ollero A, Suarez A, Marredo JM, Cioffi G, P ̀ni ̆ka R, Hoang VD, et al. (2024) Application of Intelligent Aerial Robots to the Inspection and Maintenance of Electrical Power Lines.
6. Jain N, Bedi J, Anand A, Godara S (2024) A Transfer Learning Architecture to Detect Faulty Insulators in Powerlines. *IEEE Transactions on Power Delivery* 39: 1002–1011. https://doi.org/10.1109/TPWRD.2024.3353203
7. Ahmed F, Mohanta JC (2024) Inspection of power transmission line insulators with autonomous quadcopter and SSD network. *Sigma Journal of Engineering and Natural Sciences* 42: 621–632. https://doi.org/10.14744/sigma.2024.00059
8. Ahmed MF, Mohanta JC (2024) Autonomous Site Inspection of Power Transmission Line Insulators with Unmanned Aerial Vehicle System. *Electric Power Components and Systems*, 1–24. https://doi.org/10.1080/15325008.2024.2313588
9. Chandaliya M, Goli TS, Gao J, Kotha S (2024) UAV-Based Powerline Problem Inspection and Classification using Machine Learning Approaches. In *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, 52–59.

https://doi.org/10.1109/BigDataService62917.2024.00014
10. Siddiqui ZA, Park U (2020) A Drone Based Transmission Line Components Inspection System with Deep Learning Technique. *Energies (Basel)* 13: 3348. https://doi.org/10.3390/en13133348
11. Zhu S, Li Q, Zhao J, Zhang C, Zhao G, Li L, et al. (2024) A Deep-Learning-Based Method for Extracting an Arbitrary Number of Individual Power Lines from UAV-Mounted Laser Scanning Point Clouds. *Remote Sens (Basel)* 16: 393. https://doi.org/10.3390/rs16020393
12. Liu Y, Liu D, Huang X, Li C (2023) Insulator defect detection with deep learning: A survey. *IET Generation, Transmission & Distribution* 17: 3541–3558. https://doi.org/10.1049/gtd2.12916
13. Lu Y, Li D, Li D, Li X, Gao Q, Yu X (2024) A Lightweight Insulator Defect Detection Model Based on Drone Images. *Drones* 8: 431. https://doi.org/10.3390/drones8090431
14. Cao J, Bao W, Shang H, Yuan M, Cheng Q (2023) GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. *Remote Sens (Basel)* 15: 4932. https://doi.org/10.3390/rs15204932
15. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu X (2020) GhostNet: More Features from Cheap Operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1580–1589.
16. Zhang S, Che S, Liu Z, Zhang X (2023) A real-time and lightweight traffic sign detection method based on ghost-YOLO. *Multimedia Tools and Applications* 82: 26063–26087.
17. Zhang H, Zhang H, Mei A, Gan Z, Zhu GN (2025) SO-DETR: Leveraging Dual-Domain Features and Knowledge Distillation for Small Object Detection. In *2025 International Joint Conference on Neural Networks (IJCNN)*, 1-8.
18. Zhou W, Wang X, Fan Y, Yang Y, Wen Y, Li Y (2024) KDSMALL: A lightweight small object detection algorithm based on knowledge distillation. *Comput Commun* 219: 271-281.
19. Tingyu W, Xia S, Jiaxing L, Yue Z (2024) A Deep Learning Based Detection Method for Insulator Defects in High Voltage Transmission Lines. *International Journal of Advanced Computer Science & Applications* 15. https://doi.org/10.14569/IJACSA.2024.0151040
20. Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint arXiv:2010.04159*.
21. Muzammul M, Li X, Li X (2025) Enhancing Tiny Object Detection without Fine Tuning: Dynamic Adaptive Guided Object Inference Slicing Framework with Latest YOLO Models and RT-DETR Transformer. https://doi.org/10.21203/rs.3.rs-5780163/v1
22. Joctum A, Kandiri J (2025) YOLO-APD: Enhancing YOLOv8 for Robust Pedestrian Detection on Complex Road Geometries. *International Journal of Computer Trends and Technology* 73: 58–74. https://doi.org/10.14445/22312803/IJCTT-V73I6P108
23. Zhang Q, Zhang J, Li Y, Zhu C, Wang G (2025) ID-YOLO: A Multi-Module Optimized Algorithm for Insulator Defect Detection in Power Transmission Lines. *IEEE Trans Instrum Meas* 74: 1-11. https://doi.org/10.1109/TIM.2025.3527530
24. Li D, Lu Y, Gao Q, Li X, Yu X, Song Y (2024) LiteYOLO-ID: A Lightweight Object Detection Network for Insulator Defect Detection. *IEEE Trans Instrum Meas* 73: 1-12. https://doi.org/10.1109/TIM.2024.3418082
25. Ji CL, Yu T, Gao P, Wang F, Yuan RY (2024) YOLO-TLA: An Efficient and Lightweight Small Object Detection Model based on YOLOv5. *Journal of Real-Time Image Processing* 21: 141. https://doi.org/10.1007/s11554-024-01519-4
26. Shi Z, Hu J, Ren J, Ye H, Yuan X, Ouyang Y, et al. (2025) HS-FPN: High Frequency and Spatial Perception FPN for Tiny Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* 39: 6896-6904.

27. Mirzaei B, Nezamabadi-pour H, Raoof A, Derakhshani R (2023) Small Object Detection and Tracking: A Comprehensive Review. *Sensors* 23: 6887. https://doi.org/10.3390/s23156887
28. Ali MAM, Aly T, Raslan AT, Gheith M, Amin EA (2024) Advancing Crowd Object Detection: A Review of YOLO, CNN and ViTs Hybrid Approach. *Journal of Intelligent Learning Systems and Applications* 16: 175–221. https://doi.org/10.4236/jilsa.2024.163011
29. Sharma P, Saurav S, Singh S (2024) Object detection in power line infrastructure: A review of the challenges and solutions. *Engineering Applications of Artificial Intelligence* 130: 107781. https://doi.org/10.1016/j.engappai.2023.107781
30. Abban J, and A. Awopone A (2021) Techno-Economic and Environmental Analysis of Energy Scenarios in Ghana. *Smart Grid and Renewable Energy* 12: 81–98. https://doi.org/10.4236/sgre.2021.126006
31. Liu Z, Hao Z, Han K, Tang Y, Wang Y (2024) GhostNetV3: Exploring the Training Strategies for Compact Models. *arXiv preprint arXiv:2404.11202*.
32. Natalia P, Olga P, Alexey P (2024) Efficient Method for Fast Discrete Fourier Transform of Finite Signals with High Frequency Resolution. In *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, 1–7. https://doi.org/10.1109/DSPA60853.2024.10510030
33. Xu Y, Nakayama H (2021) DCT-based Fast Spectral Convolution for Deep Convolutional Neural Networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534135
34. Alexey P, Olga P, Natalia S (2022) 2D Discrete Fast Fourier Transform with variable parameters. In *2022 24th International Conference on Digital Signal Processing and its Applications (DSPA)*, 1-8. https://doi.org/10.1109/DSPA53304.2022.9790753
35. Lin Z, Gao Y, Sang J (2022) Investigating and Explaining the Frequency Bias in Image Classification. *arXiv preprint arXiv:2205.03154*.
36. Chen Y, Liu B, Xu Y, Wu J, Chen X, Liu P, et al. (2024) Accelerating Frequency-domain Convolutional Neural Networks Inference using FPGAs. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. https://doi.org/10.1109/ISCAS58744.2024.10558358
37. Farooq U, Yang F, Shahzadi M, Ali U, Li Z (2025) YOLOv8-IDX: Optimized Deep Learning Model for Transmission Line Insulator-Defect Detection. *Electronics (Switzerland)* 14: 1828. https://doi.org/10.3390/electronics14091828

Appendix

Appendix A. Mathematical Formulation of the FAEM Filtering Pipeline.

The intrinsic feature map is projected into the frequency domain via a 2D Real FFT (extending Eq. 8):

$$F_Y = \mathcal{F}_{rfft\,2}(Y_{intrinsic}), \tag{16}$$

Here, $F_Y \in \mathbb{C}^{B\times C_p\times H_f\times W_f}$ is a complex-valued tensor that encodes both magnitude and phase components of the frequency representation.

Step 2: Learnable Frequency Filtering

At the heart of the FAEM module lies its key innovation: A learnable frequency-domain

filtering mechanism. A compact prototype mask is defined:

$$W_{proto} \in \mathbb{R}^{1\times C_p \times h_m \times w_m},\ h_m = w_m = 16 \tag{17}$$

16×16 is used because it is a compact, efficient representation of the frequency mask, and is later upsampled to cover the full FFT spectrum size.
This is interpolated to the FFT resolution:

$$W_f = \Psi_{\text{interp}}(W_{proto})_{,\ \text{size}=(H',W_f')}, \tag{18}$$

and applied to the magnitude spectrum:

$$M' = \sigma\left(|\mathcal{F}_Y| \odot W_f\right). \tag{19}$$

Finally, the original phase $\angle\mathcal{F}_Y$ is preserved and recombined with the modified magnitude to construct the filtered complex tensor.

$$\mathcal{F}_Y' = M' \cdot e^{j\cdot\angle\mathcal{F}_Y}, \tag{20}$$

Given a complex tensor $\mathcal{F}$, a learnable prototype mask W_{proto}, and an interpolation function Ψ_{interp} gives:

$$\mathcal{M}_\Psi(\mathcal{F}) = \sigma\left(|\mathcal{F}| \odot \Psi_{inter}(W_{proto})\right) \cdot e^{j\cdot\angle(\mathcal{F})}. \tag{21}$$

Step 3: Transformation Back to the Spatial Domain

The enriched spectrum is transformed back into the spatial domain using the inverse FFT denoted as $\left(\boldsymbol{\mathcal{F}^{-1}_{irfft2}}\right)$**.** The result is a preliminary version of the ghost feature map:

$$Y_{ghost\_raw} = \sigma\left(BN\left(\mathcal{F}^{-1}_{irfft\,2}(F_Y')\right)\right),\ Size = (H', W') \tag{22}$$

This step ensures that the enhanced high-frequency cues are reintroduced as spatial feature compatible with CNN operations.

Step 4: Normalization

To stabilize learning (raw ghost features), batch normalization and SiLU activation are applied:

$$Y_{ghost} = \sigma\left(BN\left(Y_{ghost_raw}\right)\right), \tag{23}$$

This ensures numerical stability and prepares the features for integration.