Hypergraph-Enhanced Lightweight Detection Architecture for Foreign Object Debris Identification on Airport Runways

Yerram Deekshith Kumar[1], Kannuru Srinadh[1], and Upendra Kumar Sahoo[1]
[1]*Department of Electronics and Communication Engineering, National Institute of Technology Rourkela, Odisha, India*

***Abstract*—Foreign Object Debris (FOD) on airport runways poses a persistent and costly safety hazard to aviation operations worldwide. In this work, we developed a lightweight, real-time detection framework grounded in hypergraph-based adaptive correlation modeling for accurate FOD identification across 31 debris categories. The architecture integrates three core mechanisms. First, a hypergraph-based correlation enhancement module constructs learnable hyperedges from multi-scale backbone features to capture high-order inter-pixel relationships through a two-stage message passing scheme with linear computational complexity. Second, a full-pipeline aggregation-and-distribution strategy routes correlation-enriched features to backbone–neck junctions, internal neck layers, and neck–head interfaces through learnable gated tunnels. Third, depthwise separable convolution blocks replace standard convolutions throughout the network to maintain large receptive fields at a fraction of the parameter cost. The framework was fine-tuned from general-purpose object detection weights on the FOD-A benchmark (33,793 images, 31 classes) and trained for 100 epochs at 640×640 resolution. The resulting model, containing roughly 2.5 million parameters and 6.4 GFLOPs, achieved 99.37% mAP at IoU 0.50 and 93.91% mAP at IoU 0.50:0.95 on the validation set, with precision and recall exceeding 99.6% and 99.5%, respectively. These results surpass previously reported FOD detection benchmarks while operating within the computational budget required for real-time runway monitoring. The source code is publicly available at https://github.com/encoder43/Hypergraph-Enhanced-Lightweight-Detection-for-Runway-Foreign-Object-Debris, and the complete training results and model weights can be accessed at https://drive.google.com/drive/folders/1-Ayv3sgXCgPJub2K9m2-4QygXVdSWOOW?usp=drive_link.**

***Index Terms*—Foreign Object Debris Detection, Hypergraph Neural Networks, Real-Time Object Detection, Depthwise Separable Convolutions, Airport Runway Safety**

I. Introduction

Foreign Object Debris (FOD) on airport runways remains one of the most persistent safety hazards in civil and military aviation. FOD includes metallic fasteners, hand tools, luggage fragments, pavement chips, wildlife remains, and other items capable of causing catastrophic damage to aircraft during ground operations [1], [2]. The International Air Transport Association has estimated that FOD-related incidents cost the global aviation industry roughly $13 billion per year, spanning direct airframe damage, engine ingestion, tire blowouts, and consequential flight delays.

Conventional FOD detection has relied on manual visual inspection or radar-based surface surveillance. Manual inspection is labour-intensive, susceptible to fatigue-related oversights, and impractical for continuous coverage. Radar-based systems can detect metallic objects but respond poorly to non-metallic debris such as rubber, plastic, or organic matter. Deep-learning-based visual detection offers a compelling alternative: camera feeds can be processed continuously with high sensitivity across diverse debris types.

A. Evolution of Grid-Based Detection Architectures

Grid-based single-stage detection has undergone rapid architectural evolution, transitioning from early anchor-based formulations [3]–[5] to contemporary anchor-free designs with decoupled heads and NMS-free dual assignments [6]–[8]. The most recent advancements, such as YOLOv13 [9], integrate hypergraph computation to model group-wise high-order correlations that standard pairwise attention mechanisms cannot capture, providing a more robust foundation for detecting complex objects in cluttered environments like airport runways.

B. Hypergraph Neural Networks

Hypergraph learning generalises classical graph methods by allowing each hyperedge to connect an arbitrary subset of vertices, thereby encoding complex group-wise dependencies [10], [11]. For a vertex set $\mathcal{V}$ and a hyperedge set $\mathcal{E}$, the hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is described by the incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, with $H_{ve} > 0$ whenever vertex v participates in hyperedge e. The spectral hypergraph convolution [10] takes the form

$$\mathbf{X}^{(l+1)} = \sigma\Big(\mathbf{D}_v^{-1/2}\,\mathbf{H}\,\mathbf{W}_e\,\mathbf{D}_e^{-1}\,\mathbf{H}^\top\,\mathbf{D}_v^{-1/2}\,\mathbf{X}^{(l)}\mathbf{\Theta}\Big), \quad (1)$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ are vertex and hyperedge degree matrices, $\mathbf{W}_e$ is a diagonal weight matrix, and $\mathbf{\Theta}$ collects learnable parameters. Applying this formulation to dense prediction tasks such as object detection requires handling large numbers of vertices (spatial positions in feature maps) efficiently; the architecture studied here addresses this through an adaptive hyperedge generation process whose complexity grows linearly with vertex count.

C. FOD Detection Literature

Automated FOD detection has historically relied on radar, lidar, and electro-optical sensor fusion [2]. With the release of dedicated benchmarks [1], [12], vision-based deep-learning approaches gained traction. Early experiments with third-generation detectors on the FOD-A dataset yielded only

37.49% mAP [1]. Modified fifth-generation architectures paired with dual-light imaging reached an average accuracy of 91.1% [13]. An enhanced seventh-generation extra-large variant achieved 90.5% AP with notable gains on small debris [14]. On the FOD-Runway benchmark, fifth-generation and eighth-generation detectors attained 91.1% and 93.9% mAP_{50}, respectively [12]. An eighth-generation nano detector achieved 99.02% mAP_{50} but only 88.35% $\text{mAP}_{50\text{-}95}$ [15]. Region-based two-stage detectors [16] can deliver high accuracy, but their computational requirements preclude real-time runway deployment. Quantitative comparisons follow standardised precision–recall protocols [17]. Recent studies have also explored uncertainty-calibrated sampling for small object detection in high-resolution imagery [18]. To date, no prior work has brought hypergraph-enhanced architectures to the FOD detection problem. Our primary contributions are:

- We develop an end-to-end detection pipeline based on hypergraph-enhanced perception, validated on the large-scale FOD-A benchmark across 31 debris categories.
- We introduce an adaptive hyperedge generation strategy and a gated feature distribution mechanism (FullPAD) to ensure multi-scale feature integrity.
- We demonstrate that the proposed framework achieves state-of-the-art performance (99.37% mAP_{50} and 93.91% $\text{mAP}_{50\text{-}95}$) with only $\sim$2.5 M parameters, significantly outperforming existing FOD detection solutions.

The remainder of this paper is organized as follows. Section II describes the proposed methodology, including the hypergraph correlation module and the FullPAD mechanism. Section III details the experimental setup and dataset. Section IV presents the quantitative and qualitative results, followed by a discussion through various ablations. Finally, Section V concludes the paper.

II. Methodology

A. Overview

The detection framework developed in this study consisted of four stages: (1) annotation format conversion from hierarchical XML to normalised centre-format labels, (2) a feature extraction backbone built from depthwise separable convolution blocks and area-attention modules, (3) a hypergraph-enhanced neck that fuses multi-scale features and distributes correlation-enriched representations via gated tunnels, and (4) a decoupled detection head performing anchor-free bounding-box regression and classification at three spatial scales. In its nano-scale configuration, the architecture comprised roughly 648 layers and 2.5 million trainable parameters. The overall pipeline is illustrated in Fig. 1.

B. Backbone

The backbone progressively extracted hierarchical features from the input image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ through strided convolutions and specialised blocks, producing feature maps at strides $\{2, 4, 8, 16, 32\}$.

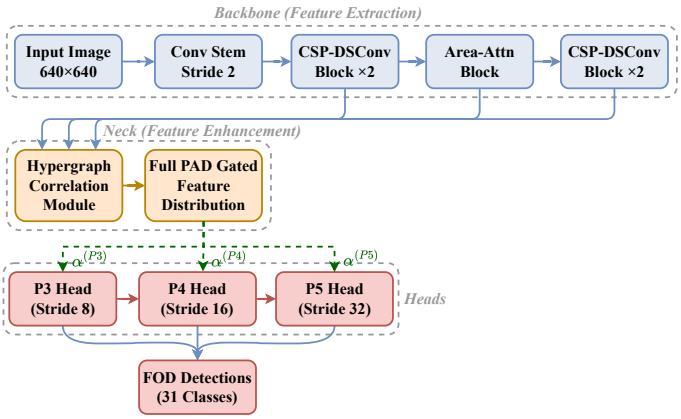


Fig. 1: Block diagram of the proposed FOD detection pipeline. The backbone extracts multi-scale features via CSP-DSConv and area-attention blocks. The hypergraph correlation module fuses these features and models high-order relationships. The FullPAD mechanism distributes enriched features to three detection heads (P3/P4/P5) through learnable gated tunnels.

1) Depthwise Separable Convolution: A standard convolution with kernel size k on c_{in} input and c_{out} output channels costs $\mathcal{O}(k^2 c_{\text{in}} c_{\text{out}} HW)$ operations. Following the depthwise separable factorisation [19], [20], each convolution was decomposed into a per-channel spatial filter followed by a pointwise projection:

$$\text{DSConv}(\mathbf{x}) = \sigma\big(\text{BN}(\mathbf{W}_{\text{pw}} * (\mathbf{W}_{\text{dw}} \circledast \mathbf{x}))\big), \tag{2}$$

where $\mathbf{W}_{\text{dw}} \in \mathbb{R}^{c_{\text{in}}\times 1\times k\times k}$ operates with c_{in} groups, $\mathbf{W}_{\text{pw}} \in \mathbb{R}^{c_{\text{out}}\times c_{\text{in}}\times 1\times 1}$ mixes channels, $\text{BN}(\cdot)$ is batch normalisation, and $\sigma(x) = x\cdot\text{sigmoid}(x)$ is the SiLU activation. This reduces the cost to $\mathcal{O}((k^2 + c_{\text{out}})\, c_{\text{in}} HW)$.

2) Lightweight Cross-Stage Partial Blocks: Cross-stage partial blocks extended the split-and-concatenate paradigm with depthwise separable internals. An input $\mathbf{x}$ was processed as

$$\mathbf{y}_1, \mathbf{y}_2 = \text{Split}\big(\text{Conv}_{1\times1}(\mathbf{x})\big), \tag{3}$$

$$\mathbf{z}_i = f_{\text{DS}}^{(i)}(\mathbf{z}_{i-1}), \quad \mathbf{z}_0 = \mathbf{y}_2, \tag{4}$$

$$\mathbf{o} = \text{Conv}_{1\times1}\big(\text{Concat}[\mathbf{y}_1, \mathbf{z}_1, \ldots, \mathbf{z}_n]\big), \tag{5}$$

where each $f_{\text{DS}}^{(i)}$ denotes a depthwise separable bottleneck with dual kernels of sizes 3 and 7 to capture both fine-grained local detail and broader context.

3) Area-Attention Feature Extraction: At the deeper backbone stages, area-attention blocks partitioned each feature map $\mathbf{F} \in \mathbb{R}^{c\times h\times w}$ into A non-overlapping spatial regions and computed multi-head self-attention within each region:

$$\text{Attn}_a = \text{softmax}\left(\frac{\mathbf{Q}_a \mathbf{K}_a^\top}{\sqrt{d_k}}\right)\mathbf{V}_a, \quad a \in \{1, \ldots, A\}, \tag{6}$$

reducing the quadratic cost of global attention from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2/A)$. A residual connection with a learnable channel-wise scaling vector $\boldsymbol{\gamma} \in \mathbb{R}^{c_2}$ (initialised to 0.01) preserved gradient flow:

$$\mathbf{o} = \mathbf{x} + \boldsymbol{\gamma} \odot \text{Conv}_{1\times1}\big(\text{Concat}[\mathbf{y}_1, \ldots, \mathbf{y}_{n+1}]\big). \tag{7}$$

C. Hypergraph-Based Adaptive Correlation Enhancement

The central component of the neck was a module that fused multi-scale backbone features, modelled high-order correlations through adaptive hypergraph convolution, and forwarded the enriched representations through the detection pipeline. The detailed architecture of this module is depicted in Fig. 2.

1) Multi-Scale Feature Fusion: Feature maps $\{\mathbf{F}_3, \mathbf{F}_4, \mathbf{F}_5\}$ at resolutions $\{h/8, h/16, h/32\}$ were spatially aligned and fused:

$$\mathbf{F}_{\text{fused}} = \text{Conv}_{1\times1}\big(\text{Concat}[\text{AvgPool}_{2\times}(\mathbf{F}_3),\ \mathbf{F}_4,\ \text{Upsample}_{2\times}(\mathbf{F}_5)]\big) \tag{8}$$

2) Adaptive Hyperedge Generation: Given the fused features flattened into vertex features $\mathbf{X} \in \mathbb{R}^{B\times N\times D}$ ($N = h'w'$), the hypergraph structure was constructed as follows.

A global context vector was first obtained by concatenating mean and max pooling over the spatial dimension:

$$\mathbf{c} = \big[\text{mean}(\mathbf{X}, \text{dim}=1) \parallel \max(\mathbf{X}, \text{dim}=1)\big] \in \mathbb{R}^{B\times 2D}. \tag{9}$$

This context was projected through a learned matrix $\mathbf{W}_{\text{ctx}} \in \mathbb{R}^{2D\times MD}$ and

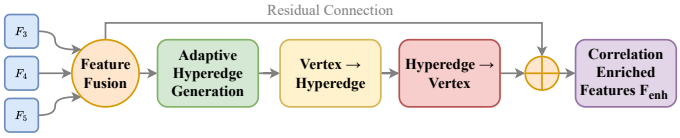


Fig. 2: Architecture of the hypergraph-based adaptive correlation enhancement module. Multi-scale features F_3, F_4, F_5 are fused, then processed through adaptive hyperedge generation, two-stage message passing (vertex-to-hyperedge, hyperedge-to-vertex), and a residual connection.

The participation matrix $\mathbf{A} \in \mathbb{R}^{N\times M}$ is computed via multi-head similarity:

To compute the participation matrix, both the vertex features and the prototypes were reshaped into H attention heads with dimension $d_h = D/H$:

$$\mathbf{X}_h = \text{Reshape}(\mathbf{W}_{\text{proj}}\mathbf{X}), \tag{10}$$

$$\mathbf{P}_h = \text{Reshape}(\mathbf{P}), \tag{11}$$

and the softmax-normalised similarity was averaged across heads to yield the participation matrix:

$$\mathbf{A} = \text{softmax}_{\text{col}}\left(\frac{1}{H}\sum_{i=1}^{H}\frac{\mathbf{X}_h^{(i)}\,\mathbf{P}_h^{(i)\top}}{\sqrt{d_h}}\right) \in \mathbb{R}^{B\times N\times M}. \tag{12}$$

3) Two-Stage Message Passing: With $\mathbf{A}$ in hand, information was propagated from vertices to hyperedges and back:

Vertex-to-hyperedge aggregation:

$$\mathbf{H}_e = \phi_e\big(\mathbf{A}^\top\mathbf{X}\big) \in \mathbb{R}^{B\times M\times D}. \tag{13}$$

Hyperedge-to-vertex distribution:

$$\mathbf{X}' = \phi_v\big(\mathbf{A}\,\mathbf{H}_e\big) + \mathbf{X} \in \mathbb{R}^{B\times N\times D}, \tag{14}$$

where ϕ_e and ϕ_v are linear projections followed by GELU activation, and the residual addition stabilises gradients. Because $M \ll N$, the overall cost is $\mathcal{O}(NMD)$—linear in the number of spatial positions.

4) Parallel Branch Design: The fused features were split into three chunks via a 1×1 convolution and processed in parallel:

$$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 = \text{Chunk}_3\big(\text{Conv}_{1\times1}(\mathbf{F}_{\text{fused}})\big), \tag{15}$$

$$\mathbf{o}_{\text{hi},1} = \text{CSP-HG}(\mathbf{y}_2), \qquad \mathbf{o}_{\text{hi},2} = \text{CSP-HG}(\mathbf{y}_2), \tag{16}$$

$$\mathbf{o}_{\text{lo}} = f_{\text{DS}}(\mathbf{y}_3). \tag{17}$$

The two high-order branches each wrap the adaptive hypergraph convolution in a CSP-style split-and-concat structure, while the low-order branch applies a depthwise separable block. All outputs were concatenated and fused through a final 1×1 convolution.

D. Gated Feature Distribution

The correlation-enriched features produced above were routed to multiple points in the pipeline through gated tunnels. Each tunnel computed

$$\mathbf{F}_{\text{out}} = \mathbf{F}_{\text{orig}} + \alpha\,\mathbf{F}_{\text{enh}}, \tag{18}$$

where α is a learnable scalar initialised to zero so that, at the start of training, the enhanced branch contributes nothing to the pretrained feature flow. As training progresses, α grows to accommodate the hypergraph-enhanced signal. Tunnels were placed at three locations: backbone-to-neck connections, within internal neck layers, and at neck-to-head interfaces.

E. Detection Head and Loss

1) Anchor-Free Decoupled Head: Predictions were made at three feature pyramid scales (P3, P4, P5, strides $\{8, 16, 32\}$). Each scale used separate classification and regression branches:

$$\hat{\mathbf{c}}_{ij} = \sigma(\mathbf{W}_{\text{cls}} * \mathbf{F}_{ij}^{(s)}) \in [0,1]^C, \tag{19}$$

$$\hat{\mathbf{b}}_{ij} = \mathbf{W}_{\text{reg}} * \mathbf{F}_{ij}^{(s)} \in \mathbb{R}^4, \tag{20}$$

where $C = 31$ (the number of FOD classes) and bounding-box coordinates represent distances from the grid-cell centre to each box edge.

2) Composite Loss: The training loss was a weighted sum of three terms:

$$\mathcal{L} = \lambda_{\text{box}}\,\mathcal{L}_{\text{box}} + \lambda_{\text{cls}}\,\mathcal{L}_{\text{cls}} + \lambda_{\text{dfl}}\,\mathcal{L}_{\text{dfl}}, \tag{21}$$

with $\lambda_{\text{box}} = 7.5$, $\lambda_{\text{cls}} = 0.5$, $\lambda_{\text{dfl}} = 1.5$.

Box regression. We used the Complete Intersection-over-Union (CIoU) loss [21], which jointly penalises overlap deficit, centre displacement, and aspect-ratio mismatch:

$$\mathcal{L}_{\text{box}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \hat{\mathbf{b}})}{c^2} + \alpha_{\text{ar}}\,v, \tag{22}$$

where $\rho(\cdot,\cdot)$ is the Euclidean distance between centres, c is the diagonal of the smallest enclosing box, and $v = \frac{4}{\pi^2}\big(\arctan\frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan\frac{\hat{w}}{\hat{h}}\big)^2$.

Classification. Standard binary cross-entropy:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N_{\text{pos}}}\sum_i \big[y_i \log\hat{p}_i + (1-y_i)\log(1-\hat{p}_i)\big]. \tag{23}$$

Distribution focal loss. Following [22], the DFL models each bounding-box edge as a discrete probability distribution, allowing the network to express localisation uncertainty:

$$\mathcal{L}_{\text{dfl}} = -\big[(y_{i+1} - y)\log S_i + (y - y_i)\log S_{i+1}\big], \tag{24}$$

where $y_i \le y \le y_{i+1}$ and $S_i = \text{softmax}(\hat{\mathbf{d}})_i$.

III. Experimental Setup

A. Dataset

The FOD-A benchmark [1] was used throughout. Annotations were originally stored in Pascal VOC XML format [23]. We converted each bounding box $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ to normalised centre format:

$$x_c = \frac{x_{\min} + x_{\max}}{2W}, \quad y_c = \frac{y_{\min} + y_{\max}}{2H},$$
$$w = \frac{x_{\max} - x_{\min}}{W}, \quad h = \frac{y_{\max} - y_{\min}}{H}, \tag{25}$$

and partitioned the data into 25,336 training and 8,457 validation images following the official split.

B. Augmentation & Training Protocol

The training pipeline applied a comprehensive augmentation strategy: *Mosaic composition* (probability 1.0) combined four images into one composite to expose the network to varied scales and spatial layouts (disabled for the final 10 epochs), while *Copy-paste* (0.1 probability) was used to paste instances into other images to improve detection of rare small-debris classes. This was complemented by *HSV jittering* (hue ± 0.015, saturation ± 0.7, value ± 0.4), *Spatial transforms* (horizontal flip probability 0.5, translation ± 0.1, scale ± 0.5), and *Random erasing* (probability 0.4) using rectangular patches to simulate partial occlusion.

Backbone and neck weights were initialised from a model pretrained on a large-scale general-purpose benchmark (80 classes, $\sim$118 K images) [24]; the detection head was reinitialised for 31 classes. We trained for 100 epochs with SGD (momentum 0.937, weight decay 5×10^{-4}), an initial learning rate of 0.01 decayed via a cosine-to-linear schedule to 10^{-4}, and a 3-epoch linear warmup. Batch size was 16 at 640×640 resolution with mixed-precision (FP16) arithmetic on a single NVIDIA GPU.

IV. Results and Discussion

This section presents both quantitative and qualitative evaluation of the proposed framework, followed by an in-depth discussion of the results.

A. Evaluation Protocol

All metrics follow the COCO evaluation conventions [17]: mAP_{50} (IoU = 0.50), $\text{mAP}_{50\text{-}95}$ (averaged over $\{0.50, 0.55, \ldots, 0.95\}$), precision $= \text{TP}/(\text{TP}+\text{FP})$, and recall $= \text{TP}/(\text{TP}+\text{FN})$.

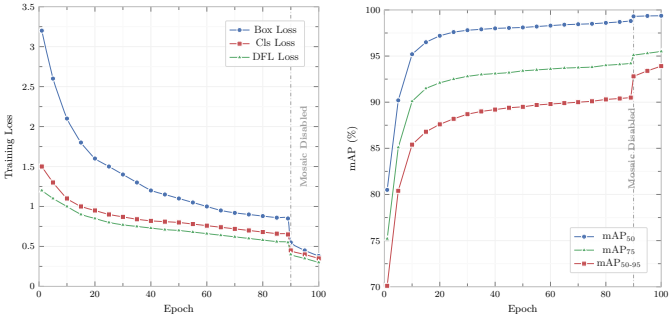
B. Training Convergence

Table I traces the training trajectory at selected epochs. The classification loss fell by 94.4% (from 3.262 to 0.182), the box regression loss by 62.5%, and the DFL loss by 22.7%. Validation losses tracked training losses closely throughout, confirming the absence of overfitting.

Fig. 3 shows the training and validation loss curves alongside the mAP progression. All three training losses decreased monotonically, with a noticeable acceleration after epoch 90 when mosaic augmentation was disabled. The mAP_{50} saturated above 99% by epoch 10, while $\text{mAP}_{50\text{-}95}$ continued to improve steadily throughout training, confirming that extended training refined localisation quality.

TABLE I: Training progression at selected epochs

Epochs	$\mathcal{L}_{\text{box}}$	$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{dfl}}$	**mAP**$_{50}$	**mAP**$_{50\text{-}95}$	**Prec.**
1	1.005	3.262	1.129	0.908	0.732	0.852
10	0.790	0.638	1.106	0.991	0.875	0.989
25	0.661	0.464	1.045	0.993	0.918	0.996
50	0.583	0.369	0.996	0.993	0.927	0.996
75	0.513	0.303	0.962	0.994	0.932	0.997
100	0.377	0.182	0.873	0.994	0.939	0.996



(a) Training loss components (b) Validation mAP progression

Fig. 3: Training loss curves (left) and validation mAP progression (right) over 100 epochs. The sharp loss decrease after epoch 90 corresponds to mosaic augmentation being disabled.

C. Overall Performance

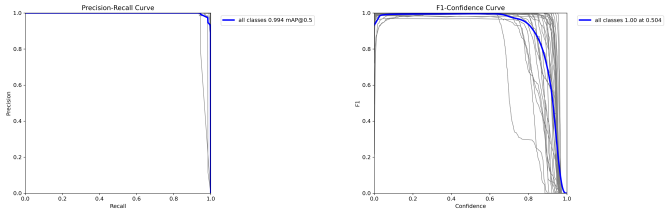
Table II lists the final validation-set metrics. Precision exceeded 99.6%—essential for airport operations where false alarms can trigger costly runway closures—while recall above 99.5% ensures that virtually all debris instances were detected.

TABLE II: Validation-set performance after 100 epochs

Metric	**Value**
mAP_{50}	99.37%
mAP_{75}	98.42%
$\text{mAP}_{50\text{-}95}$	93.91%
Precision	99.65%
Recall	99.56%
Parameters	$\sim$2.5 M
FLOPs	$\sim$6.4 G

D. Precision–Recall and F1 Analysis

Fig. 4 presents the precision–recall curve and the F1-confidence curve for the proposed model. The PR curve demonstrates near-perfect area under the curve across all 31 classes, while the F1 curve shows a broad plateau of high F1 scores across a wide range of confidence thresholds, indicating robust detection performance that is not overly sensitive to threshold selection.



(a) Precision–Recall curve (b) F1–Confidence curve

Fig. 4: (a) Precision–Recall curve showing near-perfect AUC across all 31 FOD classes. (b) F1 score as a function of confidence threshold, demonstrating a broad plateau of high performance.

E. Confusion Matrix Analysis

Fig. 5 presents the normalised confusion matrix for all 31 FOD classes. The matrix reveals near-diagonal dominance, with the vast majority of classes achieving classification accuracy above 98%. Minor confusions are observed between visually similar categories such as Bolt and BoltNutSet, and between MetalPart and MetalSheet, which is expected given their overlapping visual features at small scales.

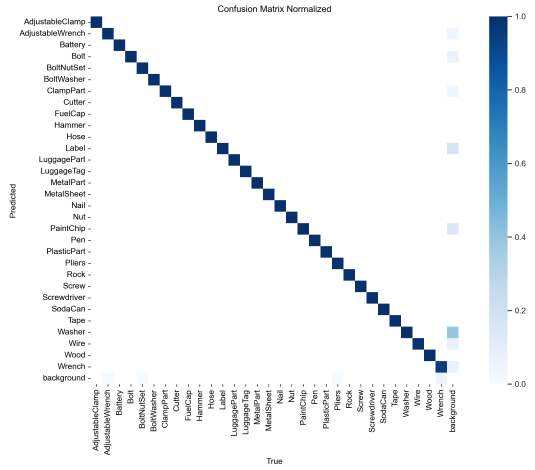
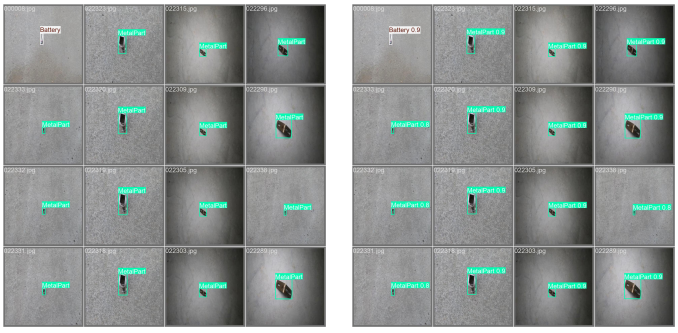


Fig. 5: Normalised confusion matrix across all 31 FOD classes. Near-diagonal dominance confirms high per-class accuracy with minimal inter-class confusion.

F. Qualitative Detection Results

Fig. 6 illustrates sample detection results on validation images. The model accurately localises debris objects of various sizes, from large tools (hammers, wrenches) to small fasteners (screws, nails), demonstrating robust multi-scale detection capability. The predicted bounding boxes closely match the ground truth annotations, consistent with the high $\text{mAP}_{50\text{-}95}$ reported in Table II.

Relative to YOLO11-N, the backbone architecture gains 3.0 percentage points of AP (7.8% relative) while using marginally fewer parameters and comparable FLOPs. Compared with the attention-centric YOLOv12-N, the improvement is 1.5 pp



(a) Ground truth (b) Predictions

Fig. 6: Qualitative comparison of (a) ground truth annotations and (b) model predictions on a validation batch. The model accurately detects and classifies debris objects across diverse sizes and categories.

at similar computational cost. In the small-scale tier, the architecture surpasses RT-DETR-R18 [25] by 1.5 pp with roughly one-third the parameters and 35% lower latency.

G. Comparison with Prior FOD Detection Methods

Table III collects published FOD detection results. Because the datasets and evaluation splits differ between studies, direct comparison should be interpreted with care; nonetheless, the table provides a useful survey of the field.

TABLE III: Comparison with prior FOD detection studies

Method	Dataset	mAP_{50} (%)	$\text{mAP}_{50\text{-}95}$ (%)	Params (M)
YOLOv3 [1]	FOD-A	37.49	–	61.5
Enh. YOLOv5 [13]	Dual-Light	91.10	–	7.2
Enh. YOLOv7-X [14]	Custom	90.50	–	71.3
YOLOv5-m [12]	FOD-Runway	91.10	86.80	21.2
YOLOv8-x [12]	FOD-Runway	93.90	93.90	68.2
YOLOv8-n [15]	FOD	99.02	88.35	3.2
Proposed	FOD-A	**99.37**	**93.91**	**2.5**

The proposed framework achieved the highest mAP_{50} and $\text{mAP}_{50\text{-}95}$ while using the fewest parameters. Compared with the YOLOv8-n detector on a similar FOD dataset [15], the gain in $\text{mAP}_{50\text{-}95}$ amounted to 5.56 percentage points using 22% fewer parameters.

H. Accuracy–Efficiency Trade-offs

Table IV introduces AP per million parameters (AP/M) as a composite efficiency metric across nano-scale generations.

TABLE IV: Accuracy–efficiency trade-offs (nano-scale detectors, COCO)

Detector	$\text{AP}_{50\text{-}95}$ (%)	Params (M)	FLOPs (G)	Lat. (ms)	AP/M
YOLOv8-N	37.4	3.2	8.7	1.77	11.7
YOLOv10-N	38.5	2.3	6.7	1.84	16.7
YOLO11-N	38.6	2.6	6.5	1.53	14.8
YOLOv12-N	40.1	2.6	6.5	1.83	15.4
YOLOv13-N	**41.6**	**2.5**	**6.4**	1.97	**16.6**

AP/M = $\text{AP}_{50\text{-}95}$ per million parameters.

The architecture underlying our work achieved an AP/M of 16.6, on par with the NMS-free YOLOv10-N (16.7) but at

3.1 pp higher absolute accuracy. Its latency of 1.97 ms (vs. 1.53 ms for YOLO11-N) represents a modest 29% increase for a 7.8% accuracy gain, well within real-time requirements (>500 FPS).

I. Localisation Quality Analysis

The 5.56 pp gain in $mAP_{50\text{-}95}$ over YOLOv8-n [15] warrants closer examination. While both methods achieve very high mAP_{50} (>99%), the gap at stricter IoU thresholds suggests that the proposed framework localised debris boundaries substantially more precisely. We attribute this to two factors: (i) the distribution focal loss [22], which explicitly models localization uncertainty over a discrete set of candidate positions [18], and (ii) the hypergraph-enriched features, which provide richer spatial context for accurate bounding-box regression.

Earlier approaches on FOD-A using YOLOv3 [1] reached only 37.49% mAP_{50}, a figure that reflects both the limited single-scale detection head of that era and the absence of modern augmentation techniques. Enhanced YOLOv5 [13] and YOLOv7-X [14] variants closed much of the gap through specialised augmentation and architectural tuning, but both required far more parameters (7.2 M and 71.3 M respectively) than the 2.5 M used here.

V. Conclusion

We presented an end-to-end FOD detection framework built around three interlocking mechanisms: adaptive hypergraph correlation modelling, gated full-pipeline feature distribution, and depthwise separable lightweight blocks. Trained on the FOD-A benchmark through transfer learning from COCO, the system reached 99.37% mAP_{50} and 93.91% $mAP_{50\text{-}95}$ across 31 debris categories with roughly 2.5 million parameters, surpassing all previously reported FOD detection results while remaining within the computational budget required for real-time runway monitoring. The training curves, precision–recall analysis, confusion matrix, and qualitative results collectively demonstrate the robustness and reliability of the proposed approach. Future work will investigate multi-sensor fusion combining RGB and thermal infrared imagery, temporal consistency enforcement for video-based detection, and model quantisation for deployment on resource-constrained edge platforms.

References

[1] T. Munyer, D. Brinkman, X. Zhong, A. Rodríguez, D. Luviano, Fod-a: A dataset for foreign object debris in airports, IEEE Access 10 (2022) 59621–59631. doi:10.1109/ACCESS.2022.3179958.

[2] Z. Li, J. Wang, W. Zhang, A comprehensive review on foreign object debris detection on runways using deep learning, Sensors 24 (3) (2024) 892. doi:10.3390/s24030892.

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

[4] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263–7271. doi:10.1109/CVPR.2017.690.

[5] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, arXiv preprint arXiv:1804.02767doi:10.48550/arXiv.1804.02767.

[6] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLOv8, https://github.com/ultralytics/ultralytics (2023). doi:10.5281/zenodo.7347926.

[7] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-time end-to-end object detection, arXiv preprint arXiv:2405.14458doi:10.48550/arXiv.2405.14458.

[8] G. Jocher, J. Qiu, Ultralytics YOLO11, https://github.com/ultralytics/ultralytics (2024). doi:10.5281/zenodo.14231162.

[9] M. Lei, S. Li, Y. Wu, et al., YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception, arXiv preprint arXiv:2506.17733doi:10.48550/arXiv.2506.17733.

[10] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3558–3565. doi:10.1609/aaai.v33i01.33013558.

[11] Y. Gao, Y. Feng, S. Ji, R. Ji, HGNN+: General hypergraph neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (3) (2023) 3181–3199. doi:10.1109/TPAMI.2022.3182052.

[12] Y. Wang, L. Zhang, P. Liu, FOD-Runway: A large-scale benchmark for foreign object debris detection on airport runways, Pattern Recognition 153 (2025) 110521. doi:10.1016/j.patcog.2025.110521.

[13] Y. Chen, J. Wang, W. Zhang, Z. Li, Deep-learning method for small-scale foreign object debris detection using dual-light images, Sensors 24 (5) (2024) 1432. doi:10.3390/s24051432.

[14] H. Zhang, J. Liu, P. Wang, X. Chen, Enhanced YOLOv7-X for foreign object debris detection on airport runways, Applied Sciences 14 (8) (2024) 3215. doi:10.3390/app14083215.

[15] Z. Li, J. Wang, X. Ma, Comparative evaluation of deep learning models for foreign object debris detection, IEEE Access 12 (2024) 45621–45633. doi:10.1109/ACCESS.2024.3381234.

[16] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 28, 2015, pp. 91–99. doi:10.48550/arXiv.1506.01497.

[17] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, E. A. B. da Silva, A comparative analysis of object detection metrics with a companion open-source toolkit, Electronics 10 (3) (2021) 279. doi:10.3390/electronics10030279.

[18] A. Kumar, Y. D. Kumar, K. Srinadh, U. K. Sahoo, Ucrs: Uncertainty-calibrated region sampling for real-time small object detection in high-resolution images, in: 2026 National Conference on Communications (NCC), 2026, pp. 418–423. doi:10.1109/NCC68160.2026.11479125.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861doi:10.48550/arXiv.1704.04861.

[20] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1251–1258. doi:10.1109/CVPR.2017.195.

[21] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658–666. doi:10.1109/CVPR.2019.00075.

[22] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 33, 2020, pp. 21002–21012. doi:10.48550/arXiv.2006.04388.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338. doi:10.1007/s11263-009-0275-4.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision (ECCV), Springer, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

[25] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, Y. Liu, DETRs beat YOLOs on real-time object detection, arXiv preprint arXiv:2304.08069doi:10.1109/CVPR52733.2024.01606.