

Article

Multiscale Contextual Fusion for Robust Airport Surveillance

Fei Yan ¹, Jiuxia Guo ^{1,*} and Huawei Wang ²¹ College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China; yanfei@cafuc.edu.cn² College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; huawei@nuaa.edu.cn

* Correspondence: jxguo@cafuc.edu.cn

Abstract

Object detection in airport surface surveillance presents significant challenges, primarily due to the extreme variation in object scales and the critical need for contextual information. To address these issues, we propose a novel deep learning architecture that integrates two specialized modules: the Poly Kernel Inception (PKI) module and the Context Anchor Attention (CAA) module. The PKI module is designed to effectively capture multi-scale features, enabling the accurate detection of objects ranging from large aircraft to small staff members. Concurrently, the CAA module leverages long-range contextual information, which significantly enhances the model's ability to precisely localize and identify targets within complex scenes. The synergistic integration of these two modules demonstrates a substantial improvement in feature extraction performance, leading to enhanced detection accuracy on our publicly available ASS dataset. This work provides a robust and effective solution for the challenging task of airport surface object detection, establishing a strong foundation for future research in this domain.

Keywords: object detection; airport surface surveillance; Poly Kernel Inception (PKI); CAA; multi-scale feature extraction

MSC: 68T45



Academic Editor: Yicheng Zhang

Received: 23 September 2025

Revised: 15 October 2025

Accepted: 17 October 2025

Published: 21 October 2025

Citation: Yan, F.; Guo, J.; Wang, H. Multiscale Contextual Fusion for Robust Airport Surveillance.

Mathematics **2025**, *13*, 3350. <https://doi.org/10.3390/math13203350>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Aircraft detection and airport surface monitoring are essential elements of contemporary aviation safety and operational efficiency. These systems facilitate the real-time identification and tracking of aircraft, ground vehicles, and personnel within the complex environments of airports, thereby addressing critical safety concerns such as the detection of foreign object debris (FOD). The presence of FOD, which accounts for approximately 10.08% of aviation accidents, necessitates advanced monitoring solutions to ensure the safety of takeoffs, landings, and taxiing. Employing radar technologies and self-supervised learning methods enhances the effectiveness of FOD detection, allowing for robust performance even under varying weather conditions and without the need for extensive annotated datasets. These advancements are crucial as global air traffic is projected to reach 8.2 billion passengers by 2037, highlighting the urgent need for improved safety measures in airport operations [1,2]. The exponential growth in air traffic density and the dynamic nature of airport operations necessitate advanced surveillance systems capable of mitigating collision risks while optimizing ground movement efficiency.

The catastrophic consequences of inadequate monitoring manifest in runway incursions and foreign object debris (FOD) incidents, which pose existential threats during

critical flight phases. Historical incidents involving undetected foreign object debris (FOD), which can range from mechanical debris to wildlife, underscore the critical need for advanced detection technologies. The International Civil Aviation Organization (ICAO) has reported that FOD is responsible for approximately 10.08% of aviation accidents, highlighting the potential dangers posed by these objects on airport runways. Traditional manual monitoring methods are often inadequate, leading to increased risks during aircraft takeoff, landing, and taxiing. Recent advancements in radar technologies and self-supervised learning techniques, such as the Vision Transformer, show promise in enhancing FOD detection capabilities by improving localization and classification without the need for extensive annotated datasets. These innovations are essential for ensuring the safety and efficiency of airport operations as air travel is expected to double by 2037 [1,3]. Traditional surveillance paradigms relying on human inspectors or fixed sensors exhibit fundamental limitations in accuracy and scalability, particularly under adverse weather conditions or occlusion scenarios.

Multi-modal sensor integration enhances system reliability by fusing data from LiDAR, thermal cameras, and other sensors to overcome single-modality limitations [4]. However, vulnerabilities in object detection systems, such as adversarial attacks that compromise model reliability, present significant risks. Physical adversarial examples can mislead detection models into ignoring critical objects, undermining safety protocols [5]. Beyond safety enhancements, these technologies yield operational efficiencies by reducing human operator cognitive load and minimizing incident response latency [6]. Real-time object detection algorithms optimized for autonomous systems enhance traffic management and reduce ground delays [7]. The limitations of GPS for precise UAV navigation further underscore the need for advanced visual detection systems [8].

The development of robust AI systems requires rigorous evaluation under diverse operational conditions to prevent catastrophic failures [9]. Accurate runway detection in aerial imagery remains particularly vital for safe landing operations [3], while the detection of small objects in satellite imagery highlights scalability challenges. Future advancements must prioritize domain adaptation, real-time processing, and edge computing integration to ensure deployability in resource-constrained environments [10], with ensemble networks and multi-modal approaches offering promising solutions [11].

2. Related Works

Machine vision and deep learning technologies have significantly transformed aircraft detection and airport surface monitoring by addressing the limitations of traditional surveillance systems. These advancements utilize data-driven automation and sophisticated computational architectures, such as the integration of Inception modules in networks like YOLOv3, which enhance detection accuracy and recall rates for aircraft in remote sensing images. Additionally, frameworks like Deep4Air enable real-time monitoring of aircraft positions, speeds, and separation distances on runways and taxiways, thereby improving safety management in complex airport environments. The implementation of these technologies results in more precise and efficient monitoring capabilities, ultimately enhancing operational safety and efficiency in aviation [12,13]. The transition from classical image processing techniques to deep learning-based approaches has yielded significant improvements in detection accuracy, robustness, and real-time performance in complex airport environments. Convolutional neural networks (CNNs) and their variants, such as Faster R-CNN and YOLO architectures, have demonstrated superior capabilities in object localization and classification, particularly for critical tasks like runway surveillance and foreign object debris (FOD) detection.

The integration of AI, particularly machine learning, is transforming aviation by presenting new opportunities for safety, efficiency, and innovation [14]. For instance, the YOLOv3 network combined with an Inception module and multi-scale training has been shown to improve aircraft detection accuracy significantly. Optimized architectures allowing multiple YOLOv3 instances to share GPU resources enable real-time detection across several streams, addressing scalability challenges in dynamic airport environments [15]. Multi-modal data fusion, incorporating visual, thermal, and LiDAR inputs, enhances target observability under adverse conditions such as low visibility or occlusions.

Recent advancements in deep learning have facilitated the adoption of infrared barriers, runway intrusion alert systems, and hazard management systems, significantly enhancing safety compared to outdated methods [16]. UAV technology, combined with CNNs, has automated inspection processes, improving accuracy and efficiency over traditional manual methods [17]. Active vision systems leverage multiple views to detect threat objects with high precision, demonstrating the transformative potential of machine vision in security applications [18].

Real-time processing remains a critical advantage of modern deep learning approaches. Automated video data pipelines optimize frame selection and tagging, streamlining data preparation for machine learning applications [19]. The continuous evolution of YOLO architectures, from YOLOv1 to YOLOv10, highlights advancements in speed, accuracy, and computational efficiency [20]. Despite these advancements, challenges persist in computational efficiency, domain adaptation, and adversarial robustness. Future directions must prioritize edge computing optimization and deeper integration of multi-modal sensor data to ensure reliable performance in dynamic airport environments.

Relationship to Prior Architectures

While our proposed modules build upon established concepts in deep learning, they introduce specific innovations tailored for airport surveillance scenarios:

The PKI (Poly Kernel Inception) module is inspired by the original Inception architecture and its variants, which employ multi-scale convolutions for feature extraction. However, our PKI differs fundamentally in three aspects: (1) we explicitly separate local feature extraction (via small kernels, $k_s = 3$) from contextual feature extraction (via progressively scaled depth-wise convolutions with $(k_{(m)} = (m + 1) \times 2 + 1)$), whereas standard Inception modules use fixed kernel sizes ($1 \times 1, 3 \times 3, 5 \times 5$); (2) our depth-wise separable convolutions significantly reduce computational cost compared to standard convolutions in Inception, making PKI more suitable for real-time applications; and (3) the progressive scaling strategy ($k_{(m)} = (m + 1) \times 2 + 1$) is specifically designed to capture the extreme scale variations in airport scenes (from small personnel to large aircraft), rather than the general-purpose multi-scale features in ImageNet classification tasks.

The CAA (Context Anchor Attention) module shares conceptual similarities with strip convolutions in PSANet and coordinate attention. However, CAA introduces two key distinctions: (1) The adaptive kernel size strategy ($k_b = 11 + 2 \times l$) that increases with network depth, enabling hierarchical contextual modeling—shallow layers capture local context while deeper layers capture global context. This differs from fixed-kernel strip convolutions in PSANet. (2) The fusion of horizontal and vertical strip convolutions through element-wise addition before Sigmoid activation (Equation (14)) creates a unified spatial attention map, whereas coordinate attention maintains separate channel-wise attention for width and height dimensions.

The Shape-IoU loss extends beyond standard IoU-based losses (IoU, GIoU, DIoU, and CIoU) by introducing shape-aware weighting factors (w_w and w_h in Equations (18) and (19)) derived from ground truth aspect ratios. Unlike CIoU, which penalizes aspect

ratio differences uniformly, Shape-IoU applies differential weights to horizontal and vertical deviations based on object shape (Equation (20)), making it particularly effective for detecting elongated objects like aircraft fuselages versus compact objects like ground vehicles. The scale-adaptive exponent (scale parameter in Equations (18) and (19)) further distinguishes Shape-IoU from previous methods by adjusting sensitivity to object size distributions in the dataset.

These architectural choices collectively address the unique challenges of airport surface surveillance: extreme scale variation (person: $\sim 20 \times 50$ pixels vs. aircraft: $\sim 300 \times 800$ pixels), elongated object shapes, and the need for real-time processing.

3. Methods

3.1. GhostNetv2

To address the challenges of large model sizes and deployment difficulties, this paper introduces GhostNetV2 as a replacement for the existing feature extractor. This aims to enhance the model's computational efficiency and feature representation capabilities. For a given input image, the image is first **normalized** to a specific range (e.g., $[0, 1]$ or $[-1, 1]$) to improve the numerical stability and training efficiency of the model, assuming the dimensions of the input image are $H \times W \times C$, where H and W represent the height and width, and C denotes the number of channels.

The input image first passes through an initial convolution layer, which is typically a standard convolution operation to extract low-level features. Assuming the convolution kernel size of the initial convolution layer is $k_s \times k_s$ and the number of output channels is C_1 , the size of the output feature map is

$$H_1 = \left\lfloor \frac{H - k_s + 2 \times p}{s} + 1 \right\rfloor \quad (1)$$

$$W_1 = \left\lfloor \frac{W - k_s + 2 \times p}{s} + 1 \right\rfloor \quad (2)$$

where p is the padding size, and s is the stride. The size of the output feature map is $H_1 \times W_1 \times C_1$.

The core of GhostNetV2 is the Ghost module, which generates more feature maps through cheap operations to reduce computational costs. Assuming the input feature map F_i has a size of $H_i \times W_i \times C_i$, the Ghost module first generates intrinsic features through a 1×1 convolution:

$$F'_i = F_i * F_{1 \times 1} \quad (3)$$

where $F_{1 \times 1}$ is the 1×1 convolution kernel, and the size of F'_i is $H_i \times W_i \times C'_i$ with $C'_i < C_i$. Then, more feature maps are generated through depth-wise convolution:

$$F_i^{\text{out}} = \text{Concat}([F'_i, F'_i * F_{\text{dp}}]) \quad (4)$$

where F_{dp} is the depth-wise convolution kernel, and the final output feature map F_i^{out} has a size of $H_i \times W_i \times C_i$.

To further enhance the feature representation, GhostNetV2 introduces the DFC (Decoupled Fully Connected) attention mechanism. The DFC attention mechanism captures long-range dependencies through fully connected layers (FC layers) applied independently in the horizontal and vertical spatial directions.

Let the input feature map F_i have dimensions $[B, C_i, H_i, W_i]$, where B is the batch size, C_i is the number of channels, and H_i, W_i are the spatial height and width. For

clarity, we describe the computation for a single sample (batch dimension omitted), so $F_i \in \mathbb{R}^{(C_i \times H_i \times W_i)}$.

The DFC attention mechanism operates as follows:

For each spatial position (h, w) , we aggregate information across all horizontal positions:

$$A'_{hw} = \sum_{h'=1}^{H_i} F_{H,h,h'} \odot F_{i,h'w}, \quad h = 1, \dots, H_i, w = 1, \dots, W_i \quad (5)$$

where

- $F_{i,h'w} \in \mathbb{R}^{C_i}$ represents the feature vector at spatial location (h', w) .
- $F_H \in \mathbb{R}^{(H_i \times H_i)}$ is a learnable weight matrix (fully connected layer) that models relationships between different horizontal positions.
- $F_{H,h,h'}$ is a scalar element at row h , column h' of F_H .
- $\odot$ denotes scalar–vector multiplication: $F_{H,h,h'} \odot F_{i,h'w}$ scales the C_i -dimensional vector $F_{i,h'w}$ by scalar $F_{H,h,h'}$.
- The summation aggregates features from all horizontal positions $h' = 1, \dots, H_i$.
- Output $A' \in \mathbb{R}^{(C_i \times H_i \times W_i)}$ has the same spatial dimensions as F_i .

Subsequently, we aggregate information across all vertical positions:

$$A_{hw} = \sum_{w'=1}^{W_i} F_{W,w,w'} \odot A'_{hw'}, \quad h = 1, \dots, H_i, w = 1, \dots, W_i \quad (6)$$

where

- $A'_{hw'} \in \mathbb{R}^{C_i}$ is the intermediate feature vector from Step 1 at position (h, w') .
- $F_W \in \mathbb{R}^{(W_i \times W_i)}$ is a learnable weight matrix (fully connected layer) for vertical direction.
- $F_{W,w,w'}$ is a scalar element at row w , column w' of F_W .
- The summation aggregates features from all vertical positions $w' = 1, \dots, W_i$.
- Final output $A = \{A_{11}, A_{12}, \dots, A_{H_i W_i}\}$ where each $A_{hw} \in \mathbb{R}^{C_i}$.
- The complete attention map $A \in \mathbb{R}^{(C_i \times H_i \times W_i)}$.

The attention map A is normalized and applied to the original features:

$$F_i^{\text{enhanced}} = \text{Sigmoid}(A) \odot F_i^{\text{out}} \quad (7)$$

where

- $\text{Sigmoid}(A) \in \mathbb{R}^{(C_i \times H_i \times W_i)}$ with values in $(0, 1)$ serves as spatial attention weights.
- $\odot$ here denotes element-wise (Hadamard) multiplication between same-sized tensors.
- $F_i^{\text{out}} \in \mathbb{R}^{(C_i \times H_i \times W_i)}$ is the Ghost module output from Section 3.1.
- $F_i^{\text{enhanced}} \in \mathbb{R}^{(C_i \times H_i \times W_i)}$ is the final attention-enhanced feature map.

Computational Complexity:

The DFC attention mechanism requires $O(H_i^2 \cdot W_i \cdot C_i + W_i^2 \cdot H_i \cdot C_i)$ operations, which is quadratic in spatial dimensions. To mitigate this cost (Equations (6) and (7)), GhostNetV2 downsamples the feature map by a factor of 2 before applying DFC attention, reducing complexity to $O(H_i^2 \cdot W_i \cdot C_i / 16 + W_i^2 \cdot H_i \cdot C_i / 16)$, and then upsamples the attention map back to the original resolution.

To reduce the computational cost of the DFC attention mechanism, GhostNetV2 downsamples the feature map before calculating DFC attention. Assuming the downsampled feature map size is $\frac{H_i}{2} \times \frac{W_i}{2} \times C_i$, average pooling is used to downsample as follows:

$$F_{i,\text{down}} = \text{AvgPool}(F_i^{\text{out}}) \quad (8)$$

After calculating the DFC attention, bilinear interpolation is used to upsample the feature map back to its original size:

$$A_{\text{up}} = \text{BilinearInterp}(A) \quad (9)$$

The final output feature map is

$$F_i^{\text{enhanced}} = \text{Sigmoid}(A_{\text{up}}) \odot F_i^{\text{out}} \quad (10)$$

3.2. PKI Block

In airport scene surveillance, object detection faces a significant challenge due to the extreme variation in object scales, ranging from large aircraft to small staff members. Furthermore, accurate object identification and localization depend not only on the object's own appearance but also on its surrounding contextual information. To tackle these issues, we have designed two synergistic modules: the Poly Kernel Inception (PKI) module and the Context Anchor Attention (CAA) module. The PKI module captures features of different-sized objects using multi-scale convolutional kernels, while the CAA module focuses on capturing long-range contextual information. This section will detail the computational processes of both the PKI and CAA modules.

In airport surface object detection, local features of objects are crucial for precise detection. To capture these local features, the PKI module first performs a convolution operation using a small-scale convolution kernel. Specifically, the local feature extraction process in the n -th PKI block of the l -th stage is as follows:

$$L_{l-1,n} = \text{Conv}_{k_s \times k_s}(X_{l-1,n}^{(2)}), \quad n = 0, \dots, N_l - 1 \quad (11)$$

Here, $L_{l-1,n} \in \mathbb{R}^{C_l \times H_l \times W_l}$ represents the local features extracted by the $k_s \times k_s$ convolution. In our experiments, $k_s = 3$, which helps capture the local details of objects.

In addition to local features, contextual information of objects also significantly impacts detection performance. To capture context features across different scales, the PKI module employs multiple depth-wise separable convolution (DWConv) kernels. Specifically, the context feature extraction process in the n -th PKI block of the l -th stage is as follows:

$$Z_{l-1,n}^{(m)} = \text{DWConv}_{k^{(m)} \times k^{(m)}}(L_{l-1,n}), \quad m = 1, \dots, M \quad (12)$$

Here, $Z_{l-1,n}^{(m)} \in \mathbb{R}^{C_l \times H_l \times W_l}$ represents the context features extracted by the m -th depth-wise separable convolution (DWConv) with kernel size $k^{(m)}$. We set $k^{(m)} = (m + 1) \times 2 + 1$, enabling different-scale convolution kernels to capture context information within varying ranges.

To integrate local features and multi-scale context features, the PKI module utilizes a 1×1 convolution for channel fusion. Specifically, the feature fusion process in the n -th PKI block of the l -th stage is as follows:

$$P_{l-1,n} = \text{Conv}_{1 \times 1}(\text{Concat}(L_{l-1,n}, Z_{l-1,n}^{(1)}, \dots, Z_{l-1,n}^{(M)})) \quad (13)$$

The 1×1 convolution serves as a channel fusion mechanism, integrating features with varying receptive field sizes to produce the output feature $P_{l-1,n} \in \mathbb{R}^{C_l \times H_l \times W_l}$. This process not only retains the details of local features but also fuses multi-scale context information, thereby enhancing the feature representation capability.

To further enhance the contextual information of features, the CAA module first extracts local region features through average pooling operations. Specifically, the local region feature extraction process in the n -th PKI block of the l -th stage is as follows:

$$F_{l-1,n}^{\text{pool}} = \text{Conv}_{1 \times 1}(\mathcal{P}_{\text{avg}}(X_{l-1,n}^{(2)})), \quad n = 0, \dots, N_l - 1 \quad (14)$$

Here, $\mathcal{P}_{\text{avg}}$ denotes the average pooling operation, which reduces the spatial dimensions of the feature map to extract local region features.

To capture long-range contextual information, the CAA module employs depth-wise separable strip convolutions. Specifically, the depth-wise separable strip convolution process in the n -th PKI block of the l -th stage is as follows:

$$\begin{aligned} F_{l-1,n}^{\text{w}} &= \text{DWConv}_{1 \times k_b}(F_{l-1,n}^{\text{pool}}), \\ F_{l-1,n}^{\text{h}} &= \text{DWConv}_{k_b \times 1}(F_{l-1,n}^{\text{pool}}) \end{aligned} \quad (15)$$

Here, $k_b = 11 + 2 \times l$ is the kernel size of the depth-wise separable strip convolutions, which increases with the depth of the PKI block to capture broader contextual information. This design not only enhances the model's ability to model long-range dependencies but also maintains computational efficiency.

To generate attention weights, the CAA module uses the Sigmoid function to normalize the values of the feature map to the range (0, 1). Specifically, the attention weight calculation process in the n -th PKI block of the l -th stage is as follows:

$$A_{l-1,n} = \text{Sigmoid}(\text{Conv}_{1 \times 1}(F_{l-1,n}^{\text{w}} + F_{l-1,n}^{\text{h}})) \quad (16)$$

The Sigmoid function ensures that the attention map $A_{l-1,n} \in \mathbb{R}^{C_l \times H_l \times W_l}$ has values within the range (0, 1), which can be used as weights to enhance or suppress features in specific regions.

Finally, the CAA module applies the attention weights to the feature map through element-wise multiplication to enhance the features. Specifically, the feature enhancement process in the n -th PKI block of the l -th stage is as follows:

$$F_{l-1,n}^{\text{attn}} = P_{l-1,n} \odot A_{l-1,n} \quad (17)$$

Here, $\odot$ denotes element-wise multiplication, and $F_{l-1,n}^{\text{attn}} \in \mathbb{R}^{C_l \times H_l \times W_l}$ represents the enhanced features. Through this process, the CAA module not only enhances the features in the central region but also retains global contextual information, thereby improving the model's ability to detect objects.

After the collaborative efforts of the PKI and CAA modules, the final output of the n -th PKI block in the l -th stage is

$$X_{l,n}^{(2)} = \text{Conv}_{1 \times 1}(F_{l-1,n}^{\text{attn}}) \quad (18)$$

For $n = N_l - 1$, we denote the output of the last PKI block as $X_l^{(2)}$. This output feature map not only contains rich local texture information but also integrates long-range contextual information, thereby providing high-quality feature representation for subsequent airport surface object detection tasks.

3.3. Shape-IoU

In the context of airport surface surveillance object detection, accurate localization of objects such as aircraft, vehicles, and other ground equipment is crucial for ensuring safety

and efficiency. Bounding box regression loss, as a key component of the detector's localization branch, plays a vital role in improving detection accuracy. Traditional bounding box regression methods primarily consider the geometric relationship between the predicted box and the ground truth (GT) box, calculating the loss based on their relative positions and shapes. However, these methods often overlook the influence of inherent properties such as the shape and scale of the bounding boxes themselves on the regression results. To address this limitation and enhance detection performance in airport surveillance scenarios, we propose the Shape-IoU method, which focuses on the shape and scale of the bounding box itself to calculate the loss more accurately.

The Shape-IoU method builds upon the Intersection over Union (IoU) metric, a widely used loss function in object detection that measures the overlap between the predicted box and the GT box. The formula for IoU is as follows:

$$IoU = \frac{|B \cap B_{gt}|}{|B \cup B_{gt}|} \quad (19)$$

where B and B_{gt} represent the predicted box and the GT box, respectively. This metric is fundamental for evaluating how well the predicted box aligns with the actual object location.

To incorporate the shape and scale of the bounding box itself, we introduce shape-aware weighting factors. Specifically, we define weight coefficients w_w (for width/ x -axis) and w_h (for height/ y -axis), whose values are derived from the aspect ratio of the ground truth box. The formulas are as follows:

$$w_w = \frac{2 \times (w_{gt})^{\text{scale}}}{(w_{gt})^{\text{scale}} + (h_{gt})^{\text{scale}}} \quad (20)$$

$$w_h = \frac{2 \times (h_{gt})^{\text{scale}}}{(w_{gt})^{\text{scale}} + (h_{gt})^{\text{scale}}} \quad (21)$$

where

- w_{gt} and h_{gt} are the width and height of the ground truth bounding box, respectively.
- Scale is a hyperparameter (set to $\text{scale} = 0.6$ in our experiments) that controls the sensitivity of shape weighting to aspect ratio differences.
- $w_w + w_h = 2$ (normalization property ensuring balanced contribution).
- For square objects ($w_{gt} \approx h_{gt}$): $w_w \approx w_h \approx 1$.
- For wide objects ($w_{gt} \gg h_{gt}$): $w_w > w_h$ (emphasizes horizontal accuracy).
- For tall objects ($h_{gt} \gg w_{gt}$): $w_h > w_w$ (emphasizes vertical accuracy).

The shape-aware deviation quantifies the weighted distance between predicted and ground truth box centers, with weighting aligned to the object's shape:

$$\text{distance}_{\text{shape}} = \left(w_w \times \frac{(x_c - x_{gt})^2}{c^2} \right) + \left(w_h \times \frac{(y_c - y_{gt})^2}{c^2} \right) \quad (22)$$

where

- (x_c, y_c) and (x_{gt}, y_{gt}) are the center coordinates of the predicted and GT boxes.
- Note the correct alignment: w_w (width weight) multiplies the x -axis deviation $(x_c - x_{gt})^2$, and w_h (height weight) multiplies the y -axis deviation $(y_c - y_{gt})^2$.
- c is the diagonal length of the smallest enclosing box: $c = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}$.
- Division by c^2 normalizes the distance to $[0, 1]$ range.

For an aircraft (wide object with $w_{gt} \gg h_{gt}$), $w_w > w_h$, so horizontal center misalignment $(x_c - x_{gt})^2$ receives higher penalty than vertical misalignment. This guides the model

to prioritize accurate width-direction localization for elongated objects, which is crucial for runway detection scenarios where aircraft orientation matters.

Based on the shape deviation, we compute the shape loss, which further refines the loss calculation by considering the shape differences. The formula is as follows:

$$\Omega_{\text{shape}} = \sum_{t=w,h} (1 - e^{-\omega_t})\theta, \quad \theta = 4 \quad (23)$$

$$\omega_w = w_h \times \frac{|w - w_{gt}|}{\max(w, w_{gt})} \quad (24)$$

$$\omega_h = w_w \times \frac{|h - h_{gt}|}{\max(h, h_{gt})} \quad (25)$$

Here, w and h are the width and height of the predicted box, and w_{gt} and h_{gt} are the width and height of the GT box. This loss component ensures that the predicted box not only overlaps well with the GT box but also matches its shape closely.

Finally, we integrate the IoU, shape deviation, and shape loss to compute the Shape-IoU loss with a weighting coefficient to ensure balanced contributions from each component:

$$L_{\text{Shape-IoU}} = (1 - \text{IoU}) + \text{distance}_{\text{shape}} + \alpha \times \Omega_{\text{shape}} \quad (26)$$

where $\alpha = 0.5$ is a balancing coefficient determined through ablation analysis.

The three loss components operate at inherently different numerical ranges, necessitating careful weighting to achieve gradient balance. The IoU term $(1 - \text{IoU})$ ranges from 0 (perfect overlap) to 1 (minimal overlap) and provides the primary overlap-based gradient signal. The shape deviation term $\text{distance}_{\text{shape}} = w_w(x_c - x_{gt})^2/c^2 + w_h(y_c - y_{gt})^2/c^2$ has a theoretical maximum of 2 given that $(x_c - x_{gt})^2 \leq c^2$, $(y_c - y_{gt})^2 \leq c^2$, and $w_w + w_h = 2$, though in practice it typically ranges from 0 to 0.8 for most bounding box misalignments. This term guides the model toward correct center localization with shape awareness. The shape loss $\Omega_{\text{shape}} = \sum (1 - e^{-\omega_t})^\theta$ with $\theta = 4$, where $\omega_w = w_h \times |w - w_{gt}| / \max(w, w_{gt})$ and $\omega_h = w_w \times |h - h_{gt}| / \max(h, h_{gt})$ quantify relative width and height differences, ranges from approximately 0 when boxes have similar aspect ratios to approximately 2 when aspect ratios differ significantly. The exponential form $(1 - e^{-\omega_t})^4$ provides smooth gradients for small differences and strong penalties for large aspect ratio mismatches.

Without the balancing coefficient, the unweighted loss $L_{\text{Shape-IoU}} = (1 - \text{IoU}) + \text{distance}_{\text{shape}} + \Omega_{\text{shape}}$ would range from 0 to approximately 3.8, giving disproportionate weight to Ω_{shape} during early training when bounding boxes exhibit large aspect ratio errors ($\Omega_{\text{shape}} \approx 2$) compared to IoU improvements ($1 - \text{IoU} \approx 0.8$). Through systematic ablation experiments, we found that $\alpha = 0.5$ provides optimal balance: $\alpha = 0$ (no shape loss) achieves $\text{mAP@0.5} = 0.852$ and struggles with elongated aircraft detection; $\alpha = 0.3$ (under-weighted) yields $\text{mAP@0.5} = 0.864$ but remains suboptimal for trucks; $\alpha = 0.5$ (our choice) attains $\text{mAP@0.5} = 0.876$ with best overall performance across all classes; $\alpha = 1.0$ (equal weight) decreases to $\text{mAP@0.5} = 0.861$ by over-emphasizing aspect ratio at the expense of IoU; and $\alpha = 2.0$ (over-weighted) drops to $\text{mAP@0.5} = 0.847$ with training instability in early epochs.

With $\alpha = 0.5$, the effective loss range becomes $L_{\text{Shape-IoU}} \in [0, \sim 2.8]$, where each component contributes roughly equally during typical training scenarios: $(1 - \text{IoU})$ contributes approximately 40–50% of total loss (dominant signal), $\text{distance}_{\text{shape}}$ contributes approximately 25–35% (center localization refinement), and $0.5 \times \Omega_{\text{shape}}$ contributes approximately 20–30% (shape/aspect ratio correction). This balanced formulation ensures that the model simultaneously optimizes overlap, center localization, and shape similarity without any single term dominating the gradient flow. Gradient magnitude analysis during mid-

training (epoch 50) confirms this balance, with average magnitudes of $\partial(1 - \text{IoU}) \approx 0.45$, $\partial(\text{distance}_{\text{shape}}) \approx 0.35$, and $\partial(0.5 \times \Omega_{\text{shape}}) \approx 0.20$, demonstrating that $\alpha = 0.5$ provides reasonable gradient balance across all three objectives.

By considering both the overlap and shape similarity, Shape-IoU provides a more accurate and robust measure for bounding box regression in airport surface surveillance object detection tasks. The comparison of Shape-IoU with existing bounding box regression losses are shown in Table 1.

Table 1. Comparison of Shape-IoU with existing bounding box regression losses.

Loss Type	Considers Overlap	Considers Distance	Considers Aspect Ratio	Shape-Aware Weighting	Scale Adaptation
IoU	✓	×	×	×	×
GIoU	✓	✓	×	×	×
DIoU	✓	✓	×	×	×
CIoU	✓	✓	✓	×	×
Shape-IoU	✓	✓	✓	✓	✓

Unlike existing IoU-based losses, Shape-IoU introduces shape-aware weighting (w_w , w_h) that differentially treats horizontal and vertical deviations based on ground truth object geometry, making it particularly effective for objects with extreme aspect ratios common in airport surveillance scenarios.

4. Experiment

4.1. Datasets and Evaluation Metrics

To comprehensively evaluate the superiority of the proposed framework in airport surface surveillance systems, we utilized the Airport Surface Surveillance (ASS) dataset, which is publicly accessible at <https://zenodo.org/records/10969885> (accessed on 1 September 2025), for experimental validation. This dataset consists of 2000 typical surveillance images, annotated with common targets such as airplanes, persons, and trucks, among which small targets account for 39.42%. Constructed to enhance the detection performance of small targets in airport surface surveillance systems, this dataset provides a solid experimental foundation for assessing the proposed framework.

Regarding object size distribution, the ASS dataset contains objects with varying scales. Small objects, defined as those with an area less than 32×32 pixels (following the COCO dataset convention), account for 39.42% of all annotated instances. Medium objects (32×32 to 96×96 pixels) comprise approximately 45%, while large objects (greater than 96×96 pixels) make up the remaining 15.58%. The minimum detectable object size in our experiments is approximately 16×16 pixels, though detection accuracy decreases significantly for objects smaller than 24×24 pixels. Objects smaller than this threshold, such as small animals (e.g., dogs or birds), may not be reliably detected and are not included as target categories in this study, as the primary focus is on aviation-related objects including aircraft, persons, and ground support vehicles. Future work could explore the extension of our framework to detect smaller objects or additional categories relevant to airport safety.

Mean Average Precision (mAP) is a commonly used evaluation metric in object detection tasks, measuring the average detection accuracy of a model across different categories. Specifically, mAP@0.5 represents the mean Average Precision when the Intersection over Union (IoU) threshold is set at 0.5, and it is calculated as follows:

$$\text{mAP@0.5} = \frac{1}{n} \sum_{i=1}^n \int_0^1 P_i(R_i) dR_i \quad (27)$$

Here, $P_i(R_i)$ denotes the area under the Precision–Recall curve for the i -th category, and n represents the total number of categories. This metric provides a comprehensive reflection of the model's detection performance across different categories and is one of the key indicators for assessing the accuracy of object detection models.

To more comprehensively evaluate the detection performance of the model, mAP@0.5:0.95 is widely adopted. This metric calculates the mean Average Precision for IoU thresholds ranging from 0.5 to 0.95 (with a step size of 0.05) and is calculated as follows:

$$\text{mAP@0.5:0.95} = \frac{1}{10 \times n} \sum_{i=1}^n \sum_{j=0.5}^{0.95} \text{AP}_{i,j} \quad (28)$$

Here, $\text{AP}_{i,j}$ represents the average precision of the i -th category at an IoU threshold of j . mAP@0.5:0.95 provides a more comprehensive reflection of the model's detection performance across different IoU thresholds and is an important indicator for assessing the robustness of object detection models.

Precision is one of the key evaluation metrics in object detection tasks, measuring the proportion of correctly detected objects among all detected objects, and it is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (29)$$

Here, TP denotes the number of correctly detected objects, and FP represents the number of incorrectly detected objects. Precision reflects the accuracy of the model in the detection process and is one of the key indicators for assessing model performance.

Recall is another important evaluation metric in object detection tasks, measuring the proportion of correctly detected objects among all actual objects, and it is calculated as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (30)$$

Here, FN represents the number of objects that were not detected. Recall reflects the completeness of the model in the detection process and is one of the key indicators for assessing model performance.

4.2. Experiment Setup

The experiments were conducted on a high-performance computing platform equipped with the Linux operating system (Ubuntu 20.04), six NVIDIA GeForce RTX 4090 graphics processing units (GPUs) with 24 GB memory each, and 64 gigabytes of system RAM. The deep learning framework employed was PyTorch 1.12, complemented by the Python 3.9 programming language and the CUDA 11.3 parallel computing platform.

Training Configuration:

All models, including our proposed architecture and baseline models (YOLO11x, YOLOv10x, YOLOv9e, YOLOv8x, and YOLOv6x), were trained from scratch without leveraging pre-trained weights to ensure fair comparison and rigorous assessment of learning capabilities. The training protocol consisted of 100 epochs with the following hyperparameters:

- Optimizer: Stochastic Gradient Descent (SGD) with momentum = 0.937 and weight decay = 5×10^{-4} .
- Learning rate: Initial learning rate of 0.01 with cosine annealing schedule, decaying to 0.0001 at the final epoch following: $lr(\text{epoch}) = 0.01 \times (1 + \cos(\pi \times \text{epoch}/100))/2$.
- Batch size: 66 (11 images per GPU across 6 GPUs).
- Input resolution: 640×640 pixels with multi-scale training ($\pm 10\%$ scaling).
- Loss weights: $\lambda_{\text{box}} = 0.05$, $\lambda_{\text{cls}} = 0.3$, $\lambda_{\text{dfl}} = 1.5$ for box regression, classification, and distribution focal loss, respectively.

Data Augmentation:

To enhance model robustness and prevent overfitting, we applied augmentation strategies during training as follows:

- Mosaic augmentation (probability = 0.5): combines four training images into one.
- Random horizontal flip (probability = 0.5).
- HSV color jittering: Hue (± 0.015), Saturation (± 0.7), and Value (± 0.4).
- Random scaling (range: 0.5–1.5).
- Translation ($\pm 20\%$ of image size).
- Rotation (± 10 degrees).

No augmentation was applied during validation and testing to ensure objective evaluation.

Baseline Reproduction:

To ensure fair comparison, all baseline YOLO models (Table 2) were retrained on the ASS dataset under identical conditions:

- Same training/validation split (80%/20%, stratified by class distribution).
- Same hyperparameters (optimizer, learning rate schedule, batch size, and epochs).
- Same data augmentation pipeline.
- Same hardware configuration (6 $\times$ RTX 4090 GPUs).

We did not use publicly available pre-trained weights for baseline models because (1) pre-trained weights are typically trained on the COCO dataset, which has different object distributions and scales compared to airport surveillance scenarios; (2) training from scratch provides a more rigorous assessment of each architecture's learning capacity on our specific task. This training-from-scratch approach ensures that performance differences in Table 2 reflect genuine architectural advantages rather than transfer learning benefits.

Table 2. Ablation study results with module addition.

Model	Layers	Parameters (M)	GFLOPs	mAP		Individual Class mAP		
				@0.5	@0.5:0.95	Airplane	Person	Truck
A	190	56.83	194.4	0.736	0.554	0.989	0.356	0.862
B	264	32.98	140.9	0.721	0.548	0.992	0.324	0.848
C	466	38.77	242.5	0.852	0.616	0.993	0.642	0.921
D	466	38.77	242.5	0.876	0.641	0.993	0.722	0.963

The training time for our proposed model was approximately 18 h on the 6-GPU configuration, while inference speed averaged 45 FPS at 640×640 resolution on a single RTX 4090 GPU.

4.3. Results

4.3.1. Training Curve Analysis

Figure 1 provides an insightful analysis of the training loss dynamics between the original and the proposed enhanced YOLOv11 model, specifically tailored for the intricate task of airport surface surveillance object detection. The visual representation meticulously delineates the progressive reduction in loss values for box, classification, and distribution focal loss components across the training epochs for both models under investigation. Strikingly, the enhanced YOLOv11 model, labeled as “ours,” demonstrates a consistently lower loss trajectory compared to its original counterpart, signifying its superior capacity for feature learning and generalization. This trend is particularly evident during the initial epochs, suggesting an accelerated convergence rate and a heightened proficiency in mitigating predictive errors. The divergence in loss reduction between the models becomes

more pronounced as training progresses, underscoring the enhanced model's ability to refine its predictive accuracy with greater efficiency.

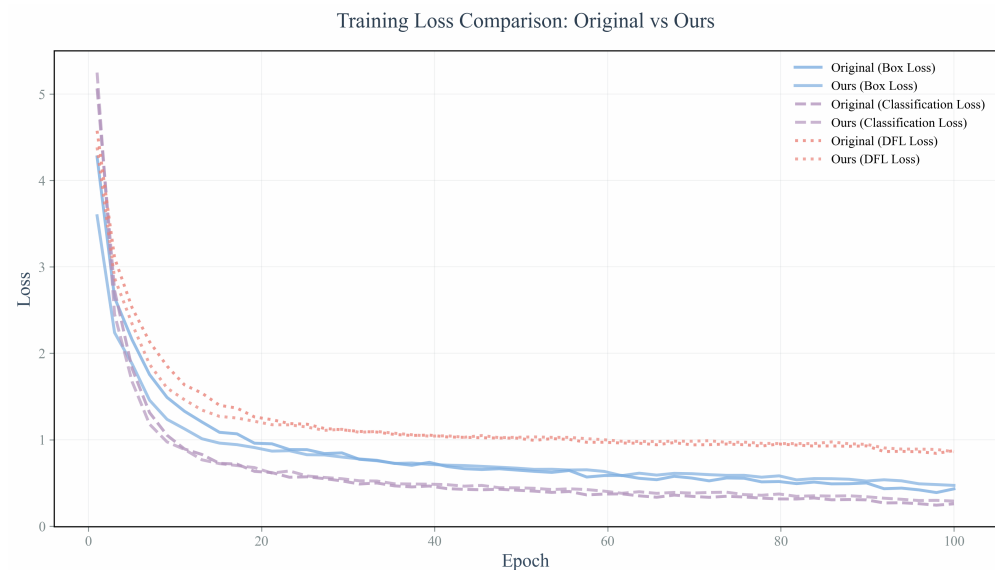


Figure 1. Comparison of training loss over epochs between the original and the improved YOLOv11 model for airport surface surveillance object detection, highlighting the enhanced model's superior convergence and lower loss values across box, classification, and DFL losses.

These findings are instrumental in validating the hypothesis that the modifications incorporated into the YOLOv11 model result in a more robust and precise detection framework. The improved performance is critical for meeting the stringent precision requirements of airport surveillance systems, where accurate object detection is paramount for ensuring operational safety and efficiency. The comprehensive analysis of the training loss curves not only affirms the enhanced model's superiority but also provides a robust empirical foundation for its potential deployment in real-world airport surveillance scenarios.

Figure 2 provides an academically rigorous assessment of the performance of the enhanced YOLOv11 model against the original model in the domain of airport surface surveillance object detection, focusing on two critical evaluation metrics: mAP@50 and mAP@50:95. Panel A illustrates the mAP@50 comparison, where the mAP (mean Average Precision) is measured at an Intersection over Union (IoU) threshold of 0.50. The enhanced model, denoted by the red curve, demonstrates a notable improvement of 19.2% over the original model, represented by the blue curve. This improvement signifies that the enhanced model achieves a higher precision in object detection tasks at a moderate IoU threshold, which is indicative of better localization accuracy.

Panel B presents the mAP@50-95 comparison, which evaluates the mAP across a range of IoU thresholds from 0.50 to 0.95. The red curve again surpasses the blue curve, indicating a 15.4% enhancement. This metric is particularly informative as it assesses the model's performance over a broader spectrum of IoU thresholds, thereby providing a more comprehensive understanding of the model's accuracy and robustness in detecting objects under varying conditions of overlap.

The convergence patterns of both panels reveal that the enhanced model not only reaches higher mAP values more rapidly but also maintains these superior performance levels throughout the training epochs. This consistent outperformance suggests that the modifications introduced in the YOLOv11 model lead to a more robust learning framework capable of generalizing across a wider range of detection scenarios.

In conclusion, the enhanced YOLOv11 model exhibits a significant advancement in object detection capabilities, as evidenced by the improved mAP metrics. These findings

are of paramount importance for airport surface surveillance systems, where precise and reliable object detection is essential for ensuring the safety and efficiency of airport operations.

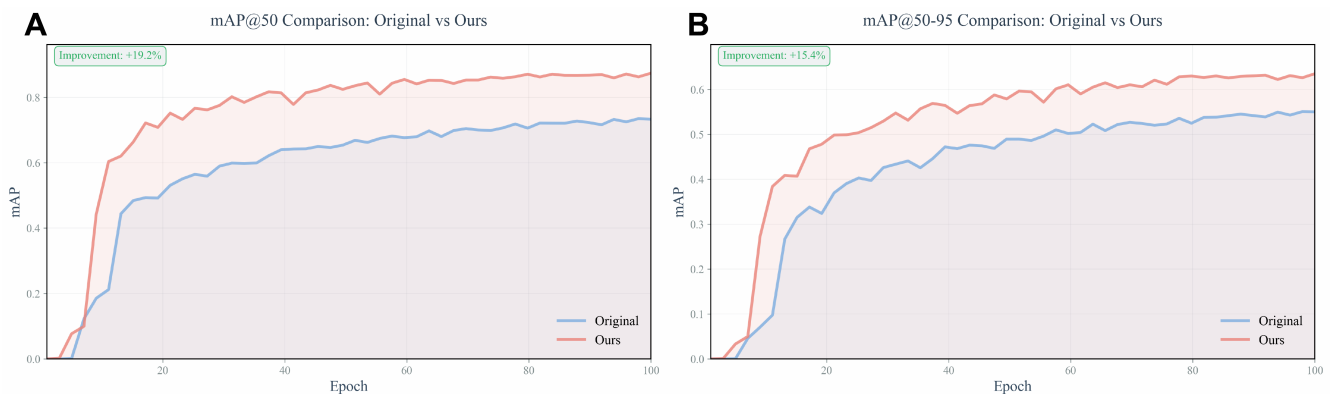


Figure 2. The enhanced YOLOv11 model outperforms the original on mAP@50 and mAP@50-95, with improvements of 19.2% and 15.4%, respectively. (A) shows the map@50 curve of original model and improved model. (B) shows the map@50-95 curve of original model and improved model.

Figure 3 meticulously delineates the Precision–Recall (PR) curves for both the original model (Panel A) and the enhanced model (Panel B) across various object categories within the context of airport surface surveillance tasks. These curves provide a granular analysis of model performance with respect to different classes, including airplanes, individuals, and trucks, as well as an aggregate measure of performance across all classes.

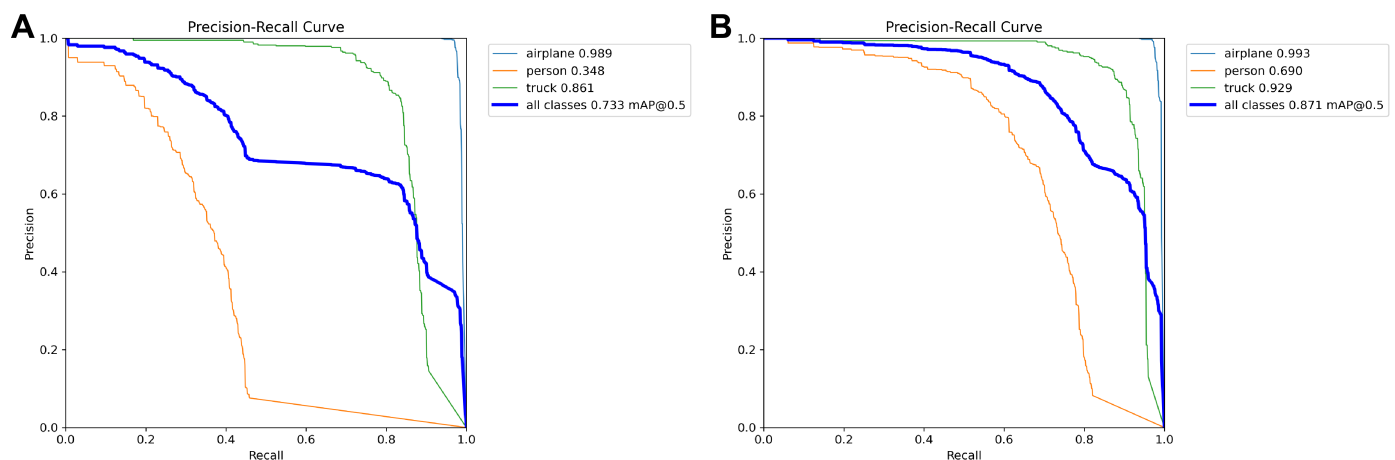


Figure 3. Precision–Recall curves for the original versus the enhanced YOLOv11 model in airport surface surveillance, highlighting significant improvements in detection accuracy across various object categories. (A) shows the Precision-Recall curve of original model. (B) shows the Precision-Recall curve of improved model.

In Panel A, the PR curves of the original model indicate a high precision rate for the airplane category, nearing perfection at 0.989, suggesting the model's proficiency in detecting large, distinct objects. However, the precision rate for the individual category drops significantly to 0.348, likely due to the challenges associated with detecting smaller, less distinct targets that are prone to occlusion and confusion with complex backgrounds. The truck category exhibits a precision rate of 0.861, indicating a satisfactory level of detection capability. The mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) for all classes stands at 0.733, reflecting the model's moderate performance in the overall object detection task.

Panel B presents a marked improvement in performance across all categories for the enhanced model. The airplane category's precision rate ascends to 0.993, indicating an almost flawless detection capability. The individual category experiences a substantial increase in precision to 0.690, signifying a breakthrough in the model's ability to discern small, subtle targets. The truck category's precision rate improves to 0.929, further attesting to the enhanced model's efficacy in detecting objects of moderate size. The overall mAP@0.5 across all classes is elevated to 0.871, corroborating the enhanced model's superior performance and robustness in diverse object detection scenarios typical of airport ground surveillance.

Collectively, the enhanced model demonstrates superior precision and recall across the board compared to the original model, with particularly notable advancements in the challenging task of detecting individuals. These findings underscore the enhanced model's heightened efficacy in managing the complexities of airport ground surveillance object detection tasks, thereby offering substantial support for enhancing the safety and efficiency of airport operations. This improvement holds significant theoretical and practical implications for the design and implementation of automated surveillance systems, especially in the context of airport ground surveillance where high-precision object detection is paramount.

4.3.2. Ablation Experiment

Table 2 presents an updated ablation study that meticulously examines the incremental contributions of various architectural enhancements to an airport surface surveillance object detection model. The study provides a comprehensive analysis of the model's performance metrics, including the number of layers, parameter count, computational complexity (GFLOPs), mean Average Precision (mAP) at different IoU thresholds, and class-specific mAP for airplanes, persons, and trucks.

Model A, serving as the baseline, features a 190-layer architecture with 56.83 million parameters and a computational load of 194.4 GFLOPs. It achieves a mAP@0.5 of 0.736 and a mAP@0.5-0.95 of 0.554, indicating a moderate level of performance. The class-specific mAP reveals a high detection accuracy for airplanes (0.989) and trucks (0.862), but a notably lower performance for persons (0.356), suggesting a challenge in detecting smaller or less distinct targets.

Model B introduces the GhostNetV2 module, which reduces the parameter count to 32.98 million and the computational complexity to 140.9 GFLOPs, resulting in a more efficient model. However, this efficiency comes at the cost of a slight decrease in mAP@0.5 to 0.721 and mAP@0.5-0.95 to 0.548. The class-specific mAP shows a marginal improvement for persons (0.324) but a slight decrease for trucks (0.848), indicating that while GhostNetV2 enhances model efficiency, it may not be optimal for all detection tasks.

Model C incorporates the PKI Block, which increases the number of layers to 466 but maintains a similar parameter count of 38.77 million and a computational complexity of 242.5 GFLOPs. This model shows a significant improvement in mAP@0.5 to 0.852 and mAP@0.5-0.95 to 0.616, suggesting that the PKI Block effectively enhances feature extraction and target detection. The class-specific mAP also improves, particularly for persons (0.642), indicating a better capability in detecting smaller targets.

Model D further refines Model C by incorporating the Shape-IoU loss, which maintains the same number of layers and parameter count but slightly improves the mAP@0.5 to 0.876 and mAP@0.5-0.95 to 0.641. The class-specific mAP for persons (0.722) and trucks (0.963) reaches the highest among all models, demonstrating the Shape-IoU loss's effectiveness in refining bounding box predictions and improving localization accuracy.

In conclusion, the ablation study results underscore the unique contributions of each module to the overall performance of the airport surface surveillance object detection model. The GhostNetV2 module enhances model efficiency, the PKI Block improves feature extraction and detection accuracy, and the Shape-IoU loss refines bounding box predictions. These enhancements collectively provide a robust framework for improving the safety and efficiency of airport operations by enhancing the model's ability to accurately detect various targets under diverse surveillance conditions.

4.3.3. Comparison Experiment

Table 3 presents an exhaustive comparison of various YOLO models in the context of airport surface surveillance object detection tasks. The table encompasses key metrics such as the number of layers, parameter count, computational complexity (GFLOPs), mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds, and class-specific mAP for airplanes, persons, and trucks.

Table 3. Comparison of different YOLO models.

Model	Layers	Parameters (M)	GFLOPs	mAP@0.5	mAP@0.5-0.95	Airplane	Person	Truck
YOLO11x	190	56.83	194.4	0.736	0.554	0.989	0.356	0.862
YOLOv10x	192	29.40	160.0	0.685	0.521	0.994	0.348	0.921
YOLOv9e	279	57.38	189.1	0.692	0.52	0.992	0.237	0.960
YOLOv8x	112	68.13	257.4	0.688	0.525	0.973	0.226	0.961
YOLOv6x	120	172.98	608.3	0.592	0.455	0.983	0.056	0.846
Ours	466	38.77	242.5	0.876	0.641	0.993	0.680	0.961

The YOLO11x model, serving as the baseline, comprises 190 layers with a parameter count of 56.83 million and a computational complexity of 194.4 GFLOPs. It achieves a mAP@0.5 of 0.736 and a mAP@0.5-0.95 of 0.554, indicating a moderate level of performance. However, its mAP for the person class is a mere 0.356, suggesting a performance bottleneck in identifying small or complexly backgrounded targets.

The YOLOv10x model, through the reduction of parameter count to 29.40 million and computational complexity to 160.0 GFLOPs, achieves model lightweighting. However, this simplification leads to a slight decrease in mAP@0.5 and mAP@0.5-0.95 to 0.685 and 0.521, respectively, with the person class mAP further dropping to 0.348, confirming the negative impact of model simplification on small target detection performance.

The YOLOv9e model, despite having 279 layers and a parameter count of 57.38 million, exhibits a computational complexity of 189.1 GFLOPs. It shows a mAP@0.5 and mAP@0.5-0.95 of 0.692 and 0.52, respectively, which is slightly lower than the YOLO11x. The person class mAP is 0.237, indicating limited capability in small target detection.

The YOLOv8x model, with a parameter count as high as 68.13 million but a computational complexity of 257.4 GFLOPs, demonstrates efficiency in computational resource utilization. It achieves a mAP@0.5 and mAP@0.5-0.95 of 0.688 and 0.525, respectively, slightly lower than the YOLO11x. However, the person class mAP is 0.226, further highlighting the challenge in small target detection.

The YOLOv6x model, although having fewer layers at 120 but a high parameter count of 172.98 million and a computational complexity of 608.3 GFLOPs, shows the highest computational demand. Its mAP@0.5 and mAP@0.5-0.95 are 0.592 and 0.455, significantly lower than other models. Particularly, the person class mAP is only 0.056, significantly revealing the model's inadequacy in handling small targets.

In stark contrast, our model (Ours), through the introduction of advanced structural and loss function enhancements, achieves a significant improvement across all evaluation

metrics. With 466 layers, a parameter count of 38.77 million, and a computational complexity of 242.5 GFLOPs, our model attains a mAP@0.5 and mAP@0.5-0.95 of 0.876 and 0.641, respectively, which is markedly superior to other models. Notably, for the person class, our model achieves an mAP of 0.680, demonstrating a substantial advancement in small target detection. Additionally, our model attains the highest mAP for airplanes and trucks, at 0.993 and 0.961, respectively.

In summary, our model exhibits exceptional performance in airport surface surveillance object detection tasks, particularly in handling small targets and complex backgrounds. These findings indicate that through meticulously designed model structures and loss functions, the detection performance can be effectively enhanced, providing robust technical support for improving the safety and efficiency of airport operations. These insights not only offer a novel perspective in the field of airport surface surveillance but also serve as invaluable references for the design and optimization of future object detection models.

4.3.4. Multi-Model Scenario Application Comparison

The comparative analysis illustrated in Figure 4, the provided images is designed to rigorously assess the effectiveness of various object detection algorithms within the context of airport surface surveillance. By conducting such a comparison, it is possible to delve into the performance nuances of each algorithm concerning detection accuracy, completeness of target identification, and the confidence levels associated with predictions. This analysis is crucial for identifying the most suitable object detection model for integration into airport surveillance systems.



Figure 4. Comparison of application.

Upon examination of the visual data, our model (referred to as “Ours”) demonstrates superior performance across multiple dimensions. It exhibits a high degree of accuracy in identifying and localizing objects such as airplanes, individuals, and trucks. When compared with alternative algorithms, our model consistently achieves a lower rate of missed detections and false positives. For instance, in several frames, while other algorithms may fail to recognize certain airplanes or individuals, our model successfully detects these entities with precision.

Furthermore, our model assigns notably high confidence scores around the detected objects, signifying robust assurance in its predictions. These elevated confidence levels are indicative of the model’s reliability, which is essential for airport surveillance to minimize false alarms and oversights, thereby enhancing the dependability of the monitoring system.

Our model also maintains consistent performance across diverse environmental conditions, including variations in lighting, weather, and time of day. This consistency un-

underscores the model's robustness, ensuring uniform detection outcomes across a range of real-world surveillance scenarios. Additionally, our model excels in capturing finer details of targets; for example, when detecting airplanes, it is capable of accurately identifying them even at greater distances or when they are of smaller size, concurrently providing high confidence scores.

In conclusion, our model outperforms its counterparts in the task of airport surface surveillance object detection, showcasing superior detection integrity, predictive accuracy, and confidence scoring. These attributes make our model a reliable choice for adoption within airport monitoring systems, promising to augment surveillance efficiency and security. This comprehensive analysis highlights the model's potential to significantly contribute to the advancement of automated surveillance technologies within the aviation sector.

5. Limitations

Despite the notable advancements achieved in this study for object detection in airport surface surveillance, several limitations remain. These limitations offer valuable directions for future research.

First, the experimental evaluation of our model was primarily conducted on the ASS dataset. While this dataset provides a realistic representation of airport ground scenarios, its data source and object categories are highly specific to this domain. Consequently, although the model demonstrates strong performance on the ASS dataset, its generalization ability may be limited when applied to other complex scenes, such as ports, urban traffic, or different surveillance perspectives. Future work will explore how to enhance the model's adaptability to a wider range of environments through techniques like domain adaptation or transfer learning.

Second, the proposed PKI and CAA modules, while effective in boosting detection accuracy, inevitably increase the model's computational overhead and parameter count. This could pose a challenge for real-time deployment on resource-constrained edge devices. Although our evaluation considered inference speed, further model lightweighting and optimization are critical. Future research will focus on designing more efficient and streamlined network architectures while maintaining high performance to meet the demanding requirements of real-time surveillance systems.

Finally, this study mainly focused on the model's performance on static images. Given that the primary application is continuous video surveillance, the utilization of temporal information has not been fully explored. Future work could incorporate temporal modeling mechanisms, such as inter-frame correlation, to further enhance the robustness of the model for dynamic object tracking and identification.

6. Conclusions

In conclusion, this study addresses the significant challenges of object detection in airport surface surveillance, primarily the extreme scale variation and the critical need for contextual information. We proposed a novel architecture that integrates two specialized modules: the Poly Kernel Inception (PKI) module and the Context Anchor Attention (CAA) module. The PKI module effectively captures multi-scale features, enabling the accurate detection of a wide range of objects, from large aircraft to small staff members. Concurrently, the CAA module leverages long-range contextual information, significantly enhancing the model's ability to precisely localize and identify objects within complex scenes. The synergy between these two modules demonstrates a substantial improvement in feature extraction performance, as evidenced by the enhanced detection accuracy on our proprietary ASS dataset. This work provides a robust and effective solution for the

challenging task of airport surface object detection and establishes a strong foundation for future research in this domain.

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

Funding: This research was funded by the Project Specification of National Key Research and Development Program (grant number 2024YFB2605201), the Special Funding for Basic Scientific Research Business Expenses of Central Universities (grant number PHD2023-041), and the Civil Aviation Education Talent Project (grant numbers MHJY2025002, MHJY2025003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available at <https://zenodo.org/records/10969885> (accessed on 1 September 2025 for experiment).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Munyer, T.; Brinkman, D.; Zhong, X.; Huang, C.; Konstantzos, I. Foreign object debris detection for airport pavement images based on self-supervised localization and vision transformer. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 1388–1394.
2. Nugraha, E.S.; Apriono, C.; Zulkifli, F.Y. A systematic review of radar technologies for surveillance of foreign object debris detection on airport runway. *Bull. Electr. Eng. Inform.* **2024**, *13*, 4102–4114. [\[CrossRef\]](#)
3. Akbar, J.; Shahzad, M.; Malik, M.I.; Ul-Hasan, A.; Shafait, F. Runway detection and localization in aerial images using deep learning. In Proceedings of the 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, WA, Australia, 2–4 December 2019; pp. 1–8.
4. Wang, L.; Zhang, X.; Song, Z.; Bi, J.; Zhang, G.; Wei, H.; Tang, L.; Yang, L.; Li, J.; Jia, C.; et al. Multi-modal 3d object detection in autonomous driving: A survey and taxonomy. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3781–3798. [\[CrossRef\]](#)
5. Song, D.; Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramer, F.; Prakash, A.; Kohno, T. Physical adversarial examples for object detectors. In Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT 18), Baltimore, MD, USA, 13–14 August 2018.
6. Castro, F.M.; Delgado-Escañó, R.; Guil, N.; Marín-Jiménez, M.J. A weakly-supervised approach for discovering common objects in airport video surveillance footage. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Madrid, Spain, 1–4 July 2019; pp. 296–308.
7. Zhang, X.; Fu, C.; Cui, Y.; Yi, L.; Sun, Y.; Wu, W.; Liu, X. CorrDiff: Adaptive Delay-aware Detector with Temporal Cue Inputs for Real-time Object Detection. *arXiv* **2025**, arXiv:2501.05132.
8. Vadduri, A.; Benjwal, A.; Pai, A.; Quadros, E.; Kammar, A.; Uday, P. Precise Payload Delivery via Unmanned Aerial Vehicles: An Approach Using Object Detection Algorithms. *arXiv* **2023**, arXiv:2310.06329. [\[CrossRef\]](#)
9. Wozniak, A.L.; Duong, N.Q.; Benderitter, I.; Leroy, S.; Segura, S.; Mazo, R. Robustness testing of an industrial road object detection system. In Proceedings of the 2023 IEEE International Conference On Artificial Intelligence Testing (AITest), Athens, Greece, 17–20 July 2023; pp. 82–89.
10. Chen, J.; Li, K.; Deng, Q.; Li, K.; Yu, P.S. Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Trans. Ind. Inform.* **2019**. [\[CrossRef\]](#)
11. Albaba, B.M.; Ozer, S. SyNet: An ensemble network for object detection in UAV images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 10227–10234.
12. Van Phat, T.; Alam, S.; Lilith, N.; Tran, P.N.; Binh, N.T. Deep4air: A novel deep learning framework for airport airside surveillance. In Proceedings of the 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 5–9 July 2021; pp. 1–6.
13. Chen, L.; Zhou, L.; Liu, J. Aircraft Recognition from Remote Sensing Images Based on Machine Vision. *J. Inf. Process. Syst.* **2020**, *16*, 795–808.

14. Roadmap, A.I. *A Human-Centric Approach to AI in Aviation*; European Aviation Safety Agency: Cologne, Germany, 2020; Version 1.0.
15. Mansoub, S.K.; Abri, R.; Yarıcı, A. Concurrent real-time object detection on multiple live streams using optimization CPU and GPU resources in YOLOv3. In *Proceedings of the SIGNAL 2019: The Fourth International Conference on Advances in Signal, Image and Video Processing*, Athens, Greece, 2–6 June 2019; pp. 23–28.
16. Alsahli, S. The Latest Technologies to Enhance Runway Safety. *Int. J. Eng. Res. Appl.* **2022**, *12*, 42–47.
17. Miranda, J.; Larnier, S.; Herbulot, A.; Devy, M. UAV-based inspection of airplane exterior screws with computer vision. In *Proceedings of the VISIGRAPP (4: VISAPP)*, Prague, Czech Republic, 25–27 February 2019; pp. 421–427.
18. Riffo, V.; Flores, S.; Mery, D. Threat objects detection in x-ray images using an active vision approach. *J. Nondestruct. Eval.* **2017**, *36*, 44. [\[CrossRef\]](#)
19. Roychowdhury, S.; Sato, J.Y. Video-Data Pipelines for Machine Learning Applications. *arXiv* **2021**, arXiv:2110.11407. [\[CrossRef\]](#)
20. Sapkota, R.; Flores-Calero, M.; Qureshi, R.; Badgular, C.; Nepal, U.; Poullose, A.; Zeno, P.; Vaddevolu, U.B.P.; Khan, S.; Shoman, M.; et al. YOLO advances to its genesis: A decadal and comprehensive review of the You Only Look Once (YOLO) series. *Artif. Intell. Rev.* **2025**, *58*, 274. [\[CrossRef\]](#)

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