Machine Vision and Applications (2026) 37:138
https://doi.org/10.1007/s00138-026-01856-3

RESEARCH

Performance optimization of YOLO-FEDER FusionNet for robust drone detection in visually complex environments

Tamara R. Lenhard[1,2,3] · Andreas Weinmann[3] · Tobias Koch[1]

Received: 26 January 2026 / Revised: 16 April 2026 / Accepted: 24 May 2026

Abstract

Drone detection in visually complex environments remains challenging due to background clutter, small object scale, and camouflage effects. While generic object detectors like YOLO exhibit strong performance in low-texture scenes, their effectiveness degrades in cluttered environments with low object-background separability. To address these limitations, this work presents an enhanced iteration of YOLO-FEDER FusionNet – a detection framework that integrates generic object detection with camouflage object detection techniques. Building upon the original architecture, the proposed iteration introduces systematic advancements in training data composition, feature fusion strategies, and backbone design. Specifically, the training process leverages large-scale, photo-realistic synthetic data, complemented by a small set of real-world samples, to enhance robustness under visually complex conditions. The contribution of intermediate multi-scale FEDER features is systematically evaluated, and detection performance is comprehensively benchmarked across multiple YOLO-based backbone configurations. Empirical results indicate that integrating intermediate FEDER features, in combination with backbone upgrades, contributes to notable performance improvements. In the most promising configuration – YOLO-FEDER FusionNet with a YOLOv8l backbone and FEDER features derived from the DWD module – these enhancements yield a reduction in FNR of up to 39.1 percentage points and an increase in mAP of up to 62.8 percentage points at an IoU threshold of 0.5, compared to the initial baseline.

Keywords Drone detection · Camouflage object detection · Feature fusion · YOLO-FEDER FusionNet

1 Introduction

The development of robust and reliable drone detection technologies has become essential for strengthening the security of infrastructures, protecting individual privacy, and maintaining regulatory compliance [1, 2]. Among prevalent detection strategies, image-based drone detection has emerged as a highly promising and effective approach, exhibiting considerable potential for practical security applications. The growing adoption of image-based detection techniques is primarily driven by the economic viability, widespread availability, and inherent compatibility of camera sensors with existing surveillance infrastructures [3]. By employing advanced *computer vision (CV)* algorithms, these techniques facilitate the timely identification of potential aerial threats and the implementation of effective countermeasures.

In drone detection applications, image analysis is typically performed using generic object detection models – most commonly YOLO-based architectures [4] such as YOLOv4 [5], YOLOv5 [6–8], YOLOv8 [9–12], or YOLOv11 [13–15]. The broad adoption of these models is largely driven by their favorable trade-off between real-time inference speed and detection accuracy (in contrast to transformer-based architectures, which are generally associated

✉ Tamara R. Lenhard
tamara.lenhard@dlr.de

Andreas Weinmann
andreas.weinmann@thws.de

Tobias Koch
tobias.koch@dlr.de

1 Institute for the Protection of Terrestrial Infrastructures, German Aerospace Center (DLR), 53757 Sankt Augustin, Germany

2 Data Science Institute, European University of Technology, 64295 Darmstadt, Germany

3 ACIDA Lab, Technical University of Applied Sciences Würzburg-Schweinfurt, 97421 Schweinfurt, Germany

Published online: 20 August 2026

with higher computational and latency overheads [16]). Moreover, generic object detection models have shown strong performance in scenarios where drones are positioned against uniform or low-clutter backgrounds – such as clear blue skies – or in environments characterized by high visual contrast between the drone and its surroundings [17]. However, their performance often degrades in visually complex scenarios with highly textured backgrounds [17–19], particularly under strong chromatic and textural similarity between the drone and the surrounding environment [8].

In line with these observations, our prior work identified considerable challenges associated with drone detection in densely vegetated environments [18]. The irregular structure of vegetation-covered backgrounds, combined with complex and dynamic lighting conditions, significantly impairs the visual detectability of drones (see Fig. 1). In addition, strong visual resemblance between drone components – such as rotor arms – and adjacent branches complicates reliable discrimination and facilitates the perceptual integration of drones into their surroundings. Consequently, these camouflage effects significantly degrade the performance of generic object detection models, adversely affecting both detection accuracy and operational robustness [8, 18]. This highlights a key limitation of generic drone detectors: their exclusive reliance on generic object-level features.

Beyond drone detection, camouflage effects also pose significant challenges in other domains – particularly in animal detection – where they have led to the development of specialized *camouflage object detection (COD)* techniques [20]. Despite their demonstrated effectiveness in animal detection and the close correspondence of challenges across both domains, the transfer of COD techniques to drone detection applications remains largely unexplored. This limited cross-domain transfer may be attributed, in part, to the inherently distinct problem formulations underlying the respective detection paradigms: While drone detection targets class-specific object localization via bounding box representations, COD focuses on category-agnostic, pixel-level foreground-background segmentation under camouflage conditions (cf. Fig. 1). Nevertheless, the fine-grained discriminative cues encoded by COD models provide a crucial source of complementary information for drone detection, directly addressing the limitations of generic object detectors in visually complex, camouflage-dominated scenes.

To leverage the potential of COD techniques and enhance the robustness of generic drone detection models, we introduced the initial version of *YOLO-FEDER FusionNet* in our earlier conference publication [21]. To the best of our knowledge, YOLO-FEDER FusionNet represents the first end-to-end *deep learning (DL)* approach that actively couples generic object detection with COD. Specifically, the framework combines YOLO-based (generic) feature encoding with FEDER [22] for COD-specific feature representation, enabling robust drone detection under both clear-visibility and camouflage conditions. This is crucial for real-world deployment in complex urban environments with high visual clutter.

Building upon our earlier work [21], this paper presents a substantially extended and refined version of YOLO-FEDER FusionNet. While the initial version of YOLO-FEDER FusionNet demonstrated strong performance – particularly compared to generic drone detection models – it also highlighted opportunities for further improvement in feature utilization,

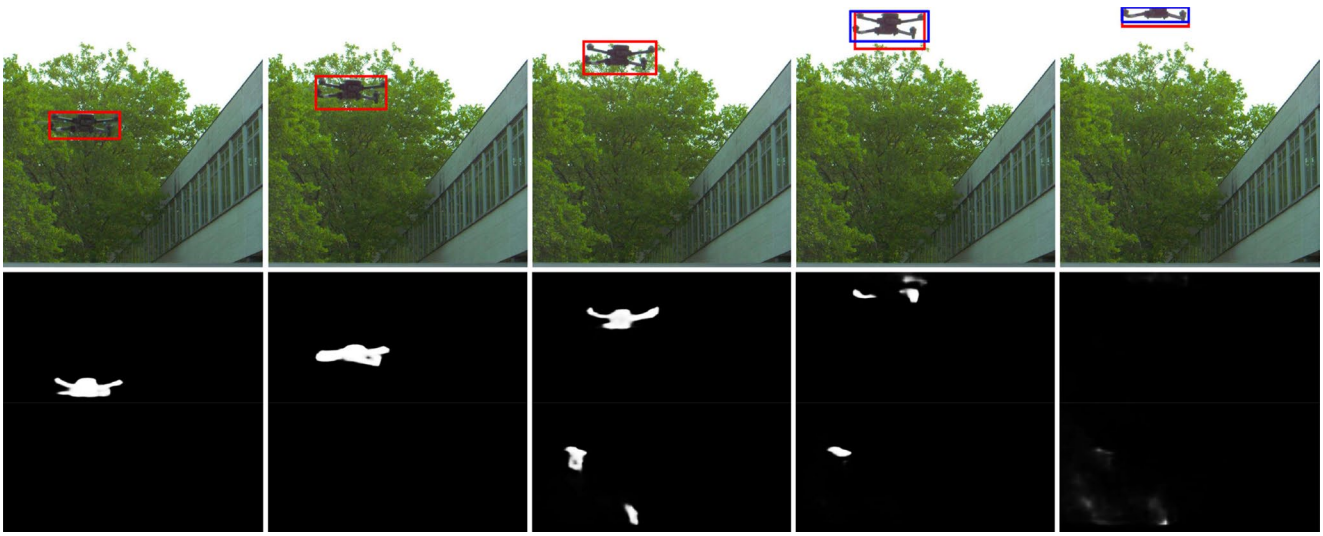

Fig. 1 Qualitative detection comparison on every 5th frame. The top row presents detections from YOLOv5l (blue) alongside YOLO-FEDER FusionNet predictions (red), while the bottom row shows FEDER (COD) outputs. YOLO-FEDER FusionNet is able to detect the drone across all evaluated frames, whereas YOLOv5l succeeds only in the final one (against a sky-dominated background). In contrast, FEDER performs well against tree-covered regions (first three frames) but degrades once the drone leaves this area

architectural design, and training strategy. Accordingly, this work introduces targeted architectural and training enhancements, including multi-scale feature fusion mechanisms (cf. Sec. 3.3), a redesigned detection head (cf. Sec. 3.4), and an improved training procedure (cf. Sec. 4.3). Beyond these refinements, we provide a comprehensive evaluation of YOLO-FEDER FusionNet and its components, leading to the following key **contributions**:

- *Training Data Optimization:* To improve the detection capabilities of YOLO-FEDER FusionNet, we strategically optimize the underlying training data by integrating large-scale, high-fidelity synthetic RGB images with a small share of real-world samples. We demonstrate the effectiveness of the optimized dataset by highlighting its performance gains over the initial YOLO-FEDER FusionNet training strategy [21].
- *Impact Analysis of Intermediate FEDER Features:* While the initial version of YOLO-FEDER FusionNet (cf. [21]) relies exclusively on the final output of the FEDER module – specifically, the binary segmentation mask – the potential contribution of intermediate FEDER feature representations remains unexplored. Accordingly, we conduct a comprehensive analysis of integrating hierarchical FEDER feature representations within YOLO-FEDER FusionNet. We assess their informational value and quantify their impact on detection performance, with particular emphasis on the multi-scale nature of FEDER's feature hierarchy.
- *Evaluation of YOLO-Based Backbone Variations:* We investigate the architectural adaptability of YOLO-FEDER FusionNet by substituting its original generic feature encoder (YOLOv5l) with more recent YOLO-based architectures, including YOLOv8, YOLOv9, and YOLOv11. A comparative analysis is performed to evaluate the resulting trade-offs between detection accuracy and computational efficiency.
- *Detection Error Assessment:* To systematically assess current performance constraints and identify opportunities for further refinement of YOLO-FEDER FusionNet, we perform a comprehensive evaluation of detection inaccuracies, explicitly focusing on the analysis and characterization of false-positive detections generated by the network.
- *Comparison to Generic Drone Detection Models:* To demonstrate the benefits of incorporating COD-inspired representations, we perform a systematic comparison between YOLO-FEDER FusionNet and state-of-the-art generic drone detection models (e.g., YOLOv5 and YOLOv8) trained under identical data, augmentation, and optimization conditions. This analysis highlights the performance gains attributable to the proposed fusion strategy, particularly in camouflage-dominated scenarios.

The paper is organized as follows: Sec. 2 reviews recent work on drone detection, camouflaged object detection, and the simulation-reality gap. Sec. 3 details the architectural design of YOLO-FEDER FusionNet. The experimental setup is outlined in Sec. 4. Experimental results are presented in Sec. 5, followed by conclusions in Sec. 6. Additional details are provided in the supplementary material (Supp. Mat.).

2 Related work

This section offers a comprehensive overview of relevant research across three key domains: image-based drone detection, camouflage object detection, and the simulation-reality gap.

2.1 Drone detection

Reliable image-based drone detection typically requires addressing a diverse set of (interconnected) challenges. These include accurately identifying small drones [6, 9, 10, 23, 24], distinguishing them from visually similar aerial entities such as birds [6], and maintaining robust detection performance under sensor imperfections [13] and in visually complex or dynamically changing environments [6, 18, 21]. Despite progress in small-drone identification and aerial object differentiation [6, 23], the development of methods explicitly designed to ensure robust performance under complex environmental conditions remains limited [6, 18]. Recent RGB-based strategies primarily focus on improving detection performance in complex scenes by systematically refining core components of established detection frameworks [6, 11]. For instance, Lv et al. [6] introduce SAG-YOLOv5s, a customized variant of YOLOv5s that integrates SimAM attention [25] and Ghost modules [26] into the bottleneck layers to improve target-specific feature representation and suppress irrelevant information during feature extraction. Building upon YOLOv8, Yu and Mo [11] introduce a GI-DGCST module – combining GSConv [27] with a dynamic group convolution shuffle transformer [28] – to strengthen fine-grained feature extraction, and a coordinate-scale attention HSFPN [29] to optimize feature selection and multi-scale feature fusion.

Other approaches address scene complexity by decomposing detection into sequential stages: background suppression [17, 19], moving-object extraction [30], and subsequent classification. Complementary strategies further extend this paradigm by integrating image-based detection into

multi-sensor fusion frameworks [31], or leveraging temporal cues via tracking to mitigate the limitations of purely camera-based systems and improve frame-level robustness in visually complex or dynamic environments [32–35]. Within this context, tracking approaches range from drone-specific techniques [32–35] to general-purpose visual object trackers, such as DeAOT [36], HQTrack [37], SAMTrack [38], and SAMURAI [39]. However, real-world deployment still relies on robust object detection for both initialization and long-term tracking stability [32].

Despite these advances, camouflage effects induced by natural surroundings – such as vegetation and tree canopies – remain a significant challenge that has not been explicitly addressed in existing methodologies (with the exception of our prior work [21]).

2.2 Camouflage object detection

Camouflage object detection is an emerging research domain focused on developing advanced methodologies for identifying objects that closely replicate the intrinsic characteristics of their surroundings [20]. These techniques are specifically designed to handle scenarios where objects exhibit minimal visual contrast and ambiguous boundaries, making accurate detection particularly challenging.

Existing COD approaches follow several methodological directions. While early works rely on modeling human visual perception to emphasize contrast and saliency cues [20, 40], more recent methods increasingly leverage the representational capacity of transformer-based architectures [41–45]. This line of work includes designs with hierarchical graph-based interaction mechanisms and dynamic token selection (e.g., HGINet [44], CTHINet [45]), as well as adaptations of large-scale foundation models (e.g., CamSAM [43]) that exploit pre-trained segmentation priors. In parallel, diffusion-based formulations model COD as a progressive mask refinement process [41, 42]. Despite their effectiveness, the associated computational complexity and reliance on iterative inference limit their suitability for latency-critical real-time applications.

Alternative COD techniques explore representation-level modeling through frequency-domain feature learning [22, 46]. A particularly promising approach in this context is the *feature decomposition and edge reconstruction (FEDER)* model by He et al. [22], which leverages frequency-aware feature decomposition to capture subtle discriminative cues, enabling improved discriminability between camouflaged objects and their surroundings. In contrast to recent transformer- and diffusion-based COD approaches, FEDER exhibits a computationally efficient architectural design, more suitable for real-time deployment.While effective in animal detection scenarios involving naturally occurring camouflage, the cross-domain generalization of COD techniques – particularly across disparate vision tasks – and their synergy with generic object detection models remain largely unexplored.

2.3 Simulation-reality gap

As real-world data collection remains resource-intensive and inherently limited in diversity, leveraging synthetically generated data has become a crucial strategy for training DL models in drone detection [18, 47–49] and other application domains [50]. The synthesis of comprehensive, application-specific datasets is typically achieved by employing physically realistic, game engine-based simulations [51] or advanced domain randomization techniques [48]. Compared to manual labeling processes, automated annotation techniques offer superior pixel-level accuracy, substantially lower associated costs, and mitigate inherent biases present in human-generated annotations. Moreover, these techniques enhance data diversity by overcoming real-world constraints, such as adverse weather conditions and privacy regulations.

Despite the benefits, the transition of synthetic-data-trained models to real-world applications remains challenging due to the simulation-reality gap. This discrepancy can lead to performance degradation, the severity of which is primarily determined by the quality and representativeness of both synthetic and real-world data. The gap between simulation and reality is commonly measured using performance indicators such as mean average precision (mAP) across varying intersection over union (IoU) thresholds [47, 48]. Common strategies for mitigating the simulation-reality gap are mixed-data training [30] and fine-tuning with real-world data [48].

3 Framework

YOLO-FEDER FusionNet effectively incorporates the capabilities of generic object detection algorithms with the distinctive strengths of COD techniques. To achieve reliable detection performance, YOLO-FEDER FusionNet leverages two backbone components for extracting meaningful features: the state-of-the-art object detection model, YOLO [4], and the camouflage object detector, FEDER [22]. Given their distinct focus and complementary outcomes (cf. Fig. 1), both backbone architectures function as an ensemble system. Thus, an input image $\mathrm{X} \in \mathbb{R}^{W \times H \times 3}$ is processed simultaneously by both backbone components (cf. Fig. 2). The extracted features are fused and refined within the network's neck, which is designed based on the architectural principles of YOLO [4]. The refined feature

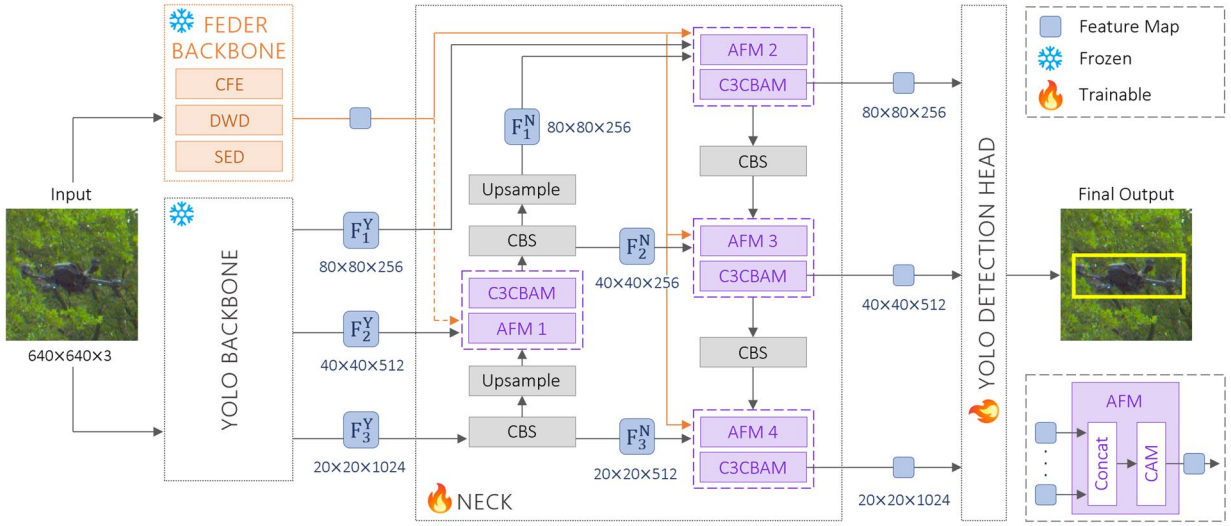

Fig. 2 Model architecture of YOLO-FEDER FusionNet. Components integrating and processing features from both backbones are highlighted in purple. Abbreviations: AFM – attention fusion module; C3CBAM – CSP bottleneck with three convolutional layers followed by channel attention; CAM – channel attention module; CBS – convolution, batch normalization, and SiLU activation; CFE – camouflage feature encoder; DWD – deep wavelet-like decomposition module; and SED – segmentation-oriented edge-assisted decoder. Feature maps F_i^Y, $i \in \{1, 2, 3\}$ originate from the YOLO backbone, while the feature maps F_i^N are obtained from neck components

Table 1 Technical overview of YOLO backbone architectures

YOLO model	No. layers	No. params (M)	GFLOPs (B)
v5l	217	26.6	69.8
v8m	142	11.8	38.8
v8l	184	19.8	86.0
v9c	293	9.0	42.8
v9e	79	30.2	117.2
v11l	238	12.7	48.0
v11x	238	28.7	107.5

maps are further processed by the network's detection head to generate (anchor-free) predictions at three distinct scales. A comprehensive overview of all YOLO-FEDER FusionNet components is provided in the subsequent sections.

3.1 YOLO backbone

A variety of pre-trained YOLO backbone architectures are employed to facilitate the extraction of generic features. Specifically, this study leverages the backbone architectures of YOLOv5, YOLOv8, YOLOv9, and YOLOv11 [4] (frequently adopted models in image-based drone detection). These backbones build upon the architectural design of CSPDarkNet53, which combines DarkNet-53 [52] with an advanced CSPNet strategy [53].

The latest YOLO iterations – specifically YOLOv5, YOLOv8, and YOLOv11 – are defined by a sequential arrangement of CBS modules, comprising convolutional layers, batch normalization, and SiLU activation functions. While variations in architectural depth and trainable parameters exist (cf. Table 1), the primary distinguishing features across these backbones are the specialized convolutional blocks: C3, C2f, and C3K2 (cf. Fig. I, Supp. Mat.). Further architectural distinctions are observed in the final layers of the backbones. While YOLOv5 and YOLOv8 terminate with a spatial pyramid pooling fusion (SPPF) module, YOLOv11's architecture incorporates both SPPF and a C2PSA module, which embeds two partial spatial attention (PSA) mechanisms. In contrast, YOLOv9 features a markedly distinct backbone architecture, built around an advanced version of the CSP-ELAN module (RepNCSPELAN) [54]. RepNCSPELAN comprises a repeated normalized cross-stage partial block (RepNCSP) and an efficient large kernel attention network (ELAN). Furthermore, YOLOv9 adopts an asymmetric downsampling technique and terminates with an SSPELAN module – a shared spatial pyramid block integrated with ELAN [55].

Regardless of variations in block design and network depth, all backbone architectures consistently output three feature maps F_i^Y, $i \in \{1, \ldots, 3\}$ with fixed dimensions of $20 \times 20 \times 1024$, $40 \times 40 \times 512$, and $80 \times 80 \times 256$ (cf. Fig. 2). These multi-scale feature maps are then processed by the YOLO-FEDER FusionNet neck for further aggregation and refinement (see Sec. 3.3).

3.2 FEDER backbone

The feature decomposition and edge reconstruction (FEDER) model, developed by He et al. [22], comprises 1,186 layers, 44.1M trainable parameters, and 200.1B (cf.

Table 2 Technical overview of FEDER components

Module	No. layers	No. params (M)	GFLOPs (B)
CFE	255	32.8	74.8
DWD	734	6.8	73.1
SED	197	4.5	52.2
Total	**1,186**	**44.1**	**200.1**

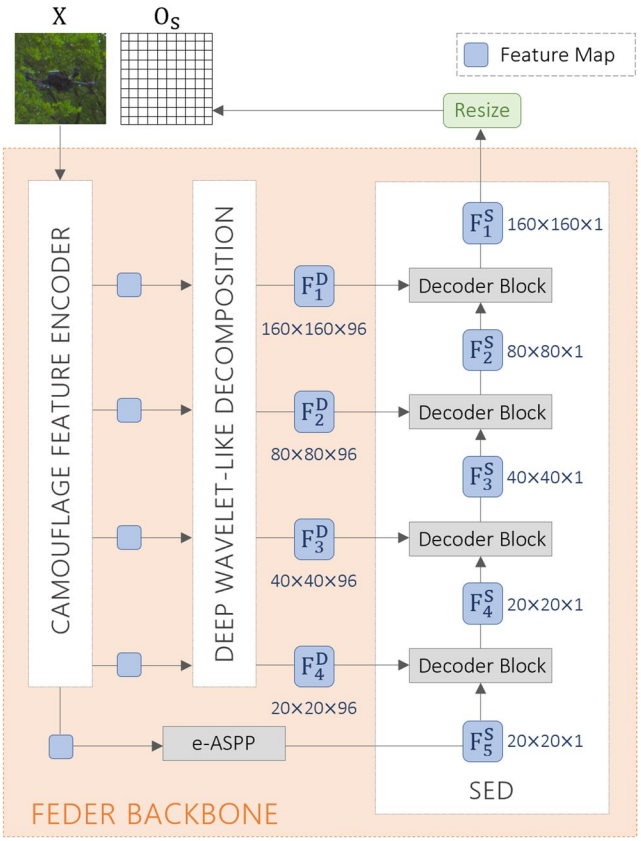

Fig. 3 FEDER backbone architecture [22], emphasizing the propagation of feature maps F_i^D, $i \in \{1, \ldots, 4\}$ and F_j^S, $j \in \{1, \ldots, 5\}$ across successive layers

Table 2). The model is built upon three key components: a camouflaged feature encoder (CFE), a deep wavelet-like decomposition (DWD) module, and a segmentation-oriented edge-assisted decoder (SED). Details of these components are provided in the following.

3.2.1 Camouflaged feature encoder

The CFE integrates a Res2Net50 architecture [56] alongside R-Net [20] to extract multi-scale feature representations from an input image $\mathrm{X} \in \mathbb{R}^{W \times H \times 3}$, where $W = H$. The extracted 64-channel feature maps are further refined by the efficient atrous spatial pyramid pooling (e-ASPP) [57] module for multi-scale contextual aggregation and the DWD module (cf. Fig. 3).

3.2.2 Deep wavelet-like decomposition module

In COD, discriminative features predominantly arise from high-frequency (HF) and low-frequency (LF) components [58]. From a signal analysis point of view, this corresponds to a spectral decomposition of visual representations, in which different frequency bands encode different semantically distinct information. While HF components encode fine-grained structural details and local discontinuities, such as texture and edge variations, LF components capture globally coherent image characteristics, including color distribution and illumination patterns. The DWD module exploits this inherent spectral distinction by applying learnable HF and LF filters, coupled with adaptive wavelet distillation [59], to systematically decompose the feature maps extracted from the CFE. By leveraging HF and LF attention modules in conjunction with a guidance-based feature aggregation technique, the DWD enhances the refinement of decomposed features, fosters inter-feature interaction, and ensures a semantically meaningful fusion strategy. The feature maps F_i^D, $i \in \{1, \ldots, 4\}$ extracted by the DWD module serve as the primary input for the segmentation-oriented edge-assisted decoder (see Fig. 3).

3.2.3 Segmentation-oriented edge-assisted decoder

The SED consists of two principal components for refined feature learning and auxiliary edge reconstruction: a reversible re-calibration segmentation (RRS) module and an ordinary differential equation (ODE)-inspired edge reconstruction (OER) module. While the RRS module seeks to amplify discriminative features in low-confidence regions via an inverse attention mechanism, the OER module focuses on enhancing edge-aware feature representation. Inspired by the dynamical systems perspective on deep neural networks (DNNs) [60], the OER module builds on the interpretation of feature propagation in residual networks as discrete approximation of a non-linear ODE. Specifically, the evolution of feature states y_t through residual layers $t \in \{0, 1, \ldots, T\}$ is described by

$$\mathrm{y}_{t+1} = \mathrm{y}_t + \Delta t \; \sigma(\mathrm{K}(\mathrm{w}_t)\mathrm{y}_t + \mathrm{b}_t) \; , \tag{1}$$

which corresponds to the first-order (forward) Euler discretization of the continuous ODE:

$$\frac{dy}{dt} = \sigma(w(t) * y(t) + b(t)), \quad y(0) = \mathrm{L}x \; , \tag{2}$$

$t \in [0, T]$. Here, $\mathrm{K}(\mathrm{w}_t)$ denotes the convolutional matrix parameterized by learnable weights w_t, b_t is the associated bias term, and $\sigma(\cdot)$ represents a non-linear activation

function. The functions $w(t)$ and $b(t)$ in Eq. 2 correspond to the continuous analogs of w_t and b_t, with $*$ denoting the continuous convolution operator. The learnable projection operator L maps the input x into the latent ODE state space. The step size $\Delta t > 0$ is fixed to one in standard DNN implementations.

To mitigate truncation errors inherent in first-order Euler discretization [22, 60], the OER module adopts a second-order, explicit Runge-Kutta (RK)-inspired scheme for more accurate feature propagation. In contrast to the standard RK formulation, which requires a manually specified trade-off parameter, the OER module incorporates a learnable gating mechanism that adaptively modulates the contribution of intermediate feature states. To further enhance stability, the module also integrates propagation techniques inspired by Hamiltonian systems [61].

The feature maps extracted by the SED are denoted by $\mathrm{F}_j^S, j \in \{1, \ldots, 4\}$ (see Fig. 3). A comprehensive overview of the architectural design and mathematical formulation of both the OER and RSS modules can be found in [22].

3.3 Neck

YOLO-FEDER FusionNet employs a neck architecture designed to facilitate the integration of multi-scale feature representations from both backbones across various hierarchical layers. Building upon the architectural principles of YOLOv5 [4], the neck structure incorporates fundamental components such as CBS blocks, C3 modules, and upsampling layers. To enhance cross-backbone feature fusion, a modified concatenation module – referred to as *attention fusion module (AFM)* – is introduced (cf. Fig. 2). The AFM combines feature concatenation with a subsequent channel attention mechanism to capture inter-channel dependencies within the aggregated feature space. The AFM-based fusion of, for instance, an intermediate YOLO-based feature map $\mathrm{F}_i^Y \in \mathbb{R}^{W \times H \times C}$, $i \in \{1, \ldots, 3\}$, and the FEDER-derived binary segmentation map $\mathrm{O}_S \in \mathbb{R}^{W \times H \times 1}$ can be formulated as follows:

$$\mathrm{AFM}(\mathrm{F}_i^Y, \mathrm{O}_S) = \mathrm{M}_C(\mathrm{F}_C) \otimes \mathrm{F}_C \ . \tag{3}$$

Here, $\mathrm{F}_C = \mathrm{Concat}(\mathrm{F}_i^Y, \mathrm{O}_S)$ denotes the concatenated feature map, while $\mathrm{M}_C(\mathrm{F}_C) \in \mathbb{R}^{1 \times 1 \times (C+1)}$ represents the corresponding channel attention map. Adopting the formulation of [62], the attention map is defined as:

$$\begin{aligned} \mathrm{M}_C(\mathrm{F}_C) = \sigma\big(& f_\theta(\mathrm{AvgPool}(\mathrm{F}_C)) \\ & + f_\theta(\mathrm{MaxPool}(\mathrm{F}_C))\big) \ , \end{aligned} \tag{4}$$

where $\mathrm{AvgPool}(\cdot)$ and $\mathrm{MaxPool}(\cdot)$ perform global average and max pooling to generate channel descriptors, followed by a shared MLP $f_\theta(\cdot)$ and sigmoid activation $\sigma(\cdot)$ to obtain channel-wise attention coefficients. The operator $\otimes$ (Eq. 3) denotes element-wise multiplication, where $\mathrm{M}_C(\mathrm{F}_C)$ is broadcast along the spatial dimensions to match the size of F_C [62]. This attention-based refinement is particularly effective for integrating information from heterogeneous sources, enabling the network to selectively emphasize the most informative features from each input.

To further enhance feature representation, the neck architecture also includes *convolutional block attention modules (CBAM)*. CBAM – originally introduced by Woo et al. [62] – is a widely employed attention mechanism that concurrently captures both spatial and channel-wise feature dependencies. Building upon the integration strategy presented in [62], where CBAM is embedded within the residual blocks of ResNet50, a comparable approach is adopted in YOLO-FEDER FusionNet. In fact, CBAM is embedded within the CSP bottleneck of the C3 module (see Fig. 2, C3CBAM) to selectively emphasize informative regions and suppress less relevant features. Given an intermediate feature map $\mathrm{F} \in \mathbb{R}^{W \times H \times C}$ derived from a preceding CBS block, the overall attention process initiated by CBAM can be described as follows:

$$\mathrm{CBAM}(\mathrm{F}) = \mathrm{M}_S(\mathbf{F}') \otimes \mathbf{F}' \tag{5}$$

with

$$\mathbf{F}' = \mathrm{M}_C(\mathrm{F}) \otimes \mathrm{F} \ . \tag{6}$$

Here, the tensor $\mathrm{M}_S(\mathrm{F}) \in \mathbb{R}^{W \times H \times 1}$ denotes the two-dimensional spatial attention map which is broadcast along the channel dimension to enable element-wise multiplication $\otimes$ with the input tensor F.

3.4 Head

YOLO-FEDER FusionNet adopts a detection head inspired by YOLOv8, integrating two stacked CBS blocks, followed by a convolutional layer and a distribution focal loss (DFL) module for enhanced bounding box regression. Classification is performed via a parallel prediction branch. The head follows an anchor-free design, eliminating the need for predefined anchors and improving generalization to objects of varying shapes and sizes. To support multi-scale detection, predictions are made at three spatial resolutions: 80×80 for small objects, 40×40 for medium objects, and 20×20 for large objects (cf. Fig. 2).

4 Experimental setup

The following sections detail the experimental setup, including the training and validation data, data pre-processing, training and evaluation protocols, and the overall experimental procedure.

4.1 Data

Given the limited availability of publicly accessible datasets and the lack of a standardized benchmark for image-based drone detection, this study integrates multiple data sources to enhance both training and evaluation. Specifically, we utilize a combination of real-world datasets – comprising self-collected and publicly available data – and synthetically generated datasets. While synthetic data is leveraged exclusively for training, real-world data constitutes the primary evaluation benchmark. Both dataset types have been introduced in our previous works [64] and [32]. Dataset statistics are summarized in Table 3, with further details provided in subsequent sections.

4.1.1 Self-captured real-world data

A ground-mounted Basler acA200-165c camera system, featuring 25 mm and 8 mm lenses, is used for real-world data collection. This configuration facilitates the acquisition of two distinct camera field-of-view from each vantage point. To ensure realistic data collection, the designated recording site closely resembles the architectural and environmental characteristics of a potential urban deployment location for drone detection systems (refer to [18] for details). The recorded RGB images are organized into two distinct datasets, R1 and R2 (cf. Table 3). Dataset R1 is characterized by urban infrastructure, with buildings comprising the majority of the background. Conversely, R2 exhibits a higher concentration of complex, highly textured elements (predominantly trees), contributing to an increased visual complexity. Both datasets are captured at a uniform resolution of 2040×1086 pixels. Data annotations are manually generated via CVAT [65]. To support reproducibility, the dataset is publicly available at [63] as part of [32].

4.1.2 Public real-world data

One of the most representative publicly accessible dataset for drone detection in a surveillance setup is the Dalian University of Technology (DUT) Anti-UAV dataset [35]. Designed for both drone detection and tracking, this dataset features a broad range of image resolutions (from 240×160 to 5616×3744 pixels) and encompasses diverse drone models, environments, lighting conditions, and weather scenarios (see [35] for details).

4.1.3 Synthetic data

The creation of synthetic training data relies on a game engine-based data generation pipeline. By leveraging the advanced capabilities of Unreal Engine (4.27 and 5.0, respectively) [66] and Colosseum [67] – a successor of AirSim [68] – the pipeline enables the systematic acquisition of automatically annotated RGB images. The pipeline's technical specifications are comprehensively detailed in [51].

Using this pipeline, two distinct synthetic datasets are generated (cf. Table 3): S1, a smaller and simpler dataset, and SynDroneVision [64], a large-scale dataset characterized by a wide variety of drone models, environments, and illumination conditions. While the generation of SynDroneVision prioritizes scalability and data diversity to reflect real-world complexities, S1 is specifically designed to capture the essential attributes of R1 and R2. The data in S1 is provided at a resolution of 2040×1080 pixels, whereas SynDroneVision features a higher resolution of 2560×1489 pixels. In addition to drone images, both datasets include a small share of background images (7-8%). For a detailed description of SynDroneVision, refer to [64]. For further information on S1, see [18].

4.2 Data pre-processing

A two-stage cropping methodology is employed to meet the model's requirement for square input images. The initial stage involves a coarse cropping strategy, guided by the precise localization of the drone's ground-truth bounding box and predefined cropping dimensions. This step ensures the

Table 3 Overview of employed synthetic and real-world datasets

Dataset	Type	Pub. Avail.	Image Count				Resolution (px)	Camera Param.		Drone Models
			train	val	test	total		pos.	focal length	
R1 [63]	real	✓	7,524	2,508	2,508	12,540	2040×1086	2	8 mm	1
R2 [63]	real	✓	3,834	1,279	1,278	6,391	2040×1086	2	25 mm	1
S1	synth.	✗	10,446	3,483	3,483	17,412	2040×1080	5	–	3
DUT Anti-UAV [35]	real	✓	5,200	2,600	2,200	10,000	_★	_▲	–	35
SynDroneVision [64]	synth.	✓	131,238	8,800	4,000	140,038	2560×1489	58	–	13

★ No uniform image resolution, variations between 240×160 to 5616×3744 pixels

▲ Variations included, but no explicit specifications

drone's retention in the subsequent cropping phase, irrespective of the cropping parameters. To enhance data variability, the second stage leverages a random cropping mechanism using distinct dimensions: 640×640 (default input size for YOLO), 1080×1080, and a dynamically determined size based on the smallest image dimension. The dynamic adaptation of cropping sizes optimizes the preservation of informational content, while maintaining flexibility across varying image resolutions. This cropping strategy generates diverse dataset representations (see Table 4).

4.3 Training specifications

YOLO-FEDER FusionNet is trained based on the following selection of augmentation techniques, hyperparameters, and a tailored loss function.

4.3.1 Augmentation techniques

To enhance data diversity and improve model robustness, we employ a combination of standard augmentation techniques. This includes random horizontal flipping with a probability of 0.5 and HSV augmentation, where hue, saturation, and brightness are dynamically adjusted within normalized ranges of ± 0.015, ± 0.7, and ± 0.4, respectively. Additionally, we incorporate a refined adaptation of mosaic augmentation, inspired by [4]. To mitigate biases in YOLO-based detection and prevent distortions in COD, we deliberately exclude letterboxing – i.e., artificial padding used to maintain aspect ratios – from mosaic augmentation. No additional blurring is applied, as the SynDroneVision dataset inherently includes blurred images, making further modifications unnecessary.

4.3.2 Hyperparameter settings

The training process is conducted over 100 epochs with a batch size of 64 and an input image size of 640×640, utilizing four Nvidia A100 GPUs. Optimization is performed using stochastic gradient descent (SGD) with an initial learning rate of 0.01 and a momentum coefficient of 0.937. To further enhance convergence efficiency, an exponential moving average is applied. All other hyperparameter settings remain consistent with those used in [4].

Table 4 Overview of datasets and cropping resolutions. "dyn." denotes a dynamically adjusted cropping size

Dataset (DS)	Cropping Resolution			DS Vers.
	640×640	1080×1080	dyn.	
R1 [63]	✓	✓	–	2
R2 [63]	✓	✓	–	2
S1	✓	✓	–	1
DUT Anti-UAV [35]	–	–	✓	1
SynDroneVision [64]	–	–	✓	1

4.3.3 Loss function

The loss function employed in the training process follows the structure of YOLOv8 and consists of three key components: the bounding box regression loss $\mathcal{L}_{\text{box}}$, the classification loss $\mathcal{L}_{\text{cls}}$, and the distribution focal loss $\mathcal{L}_{\text{dfl}}$. While $\mathcal{L}_{\text{cls}}$, formulated using *binary cross-entropy (BCE)*, ensures precise and robust classification, both $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{dfl}}$ contribute to bounding box localization. Specifically, the CIoU-based loss $\mathcal{L}_{\text{box}}$ enhances localization accuracy by optimizing spatial alignment, aspect ratio consistency, and proximity to ground truth. In contrast, $\mathcal{L}_{\text{dfl}}$ models a probability distribution over bounding box coordinates rather than directly regressing them. This is particularly effective in scenarios involving both synthetic and real-world data, where the pixel-level precision of synthetic annotations contrasts with the inherent variability and uncertainty of manually annotated real-world data. The total loss $\mathcal{L}$ is expressed as

$$\mathcal{L} = w_1 \mathcal{L}_{\text{box}} + w_2 \mathcal{L}_{\text{cls}} + w_3 \mathcal{L}_{\text{dfl}} \ , \tag{7}$$

where the non-negative weighting coefficients w_1, w_2 and w_3 significantly influence the optimization dynamics and overall model performance. While YOLOv8's default weight configuration is empirically tuned for the COCO benchmark dataset, these values may not generalize effectively across varying data domains. To ensure optimal performance for our specific application, we conducted a comprehensive ablation study, leading to the selection of $w_1 = 0.1$, $w_2 = 0.5$, and $w_3 = 1.5$. Details of the ablation study are provided in Tables I and II (Supp. Mat.).

4.4 Evaluation details

The evaluation of all trained iterations of YOLO-FEDER FusionNet (cf. Sec. 4.5) is based upon the following datasets and performance indicators.

4.4.1 Data

Performance evaluation relies exclusively on real-world data, using the datasets R1 and R2 across both cropping dimensions (cf. Tables 3 and 4), as well as the publicly available DUT Anti-UAV test set [35].

4.4.2 Metrics

Effective mitigation of unauthorized drone intrusions requires timely and accurate detection, making the

reduction of false negatives (FN) a critical factor for operational robustness. However, in practical surveillance applications, detecting a drone in every individual frame of an image sequence is not imperative, as undetected instances can often be inferred from adjacent frames, enabling a partial reconstruction of the drone's presence. Within a broader security framework, minimizing false positives (FP) is nearly as important as reducing false negatives to maintain system credibility and operational efficiency.

To provide a holistic evaluation of detection performance, we assess YOLO-FEDER FusionNet using the false negative rate (FNR) and the false discovery rate (FDR), alongside mAP at an IoU threshold of 0.5 – a standard metric in object detection. The false negative rate is defined as $\mathrm{FNR} = \mathrm{FN}/(\mathrm{FN} + \mathrm{TP})$, representing the proportion of missed detections among all ground-truth objects, while the false discovery rate, defined as $\mathrm{FDR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TP})$, measures the proportion of false detections among all predicted objects. Here, TP denotes true positives, i.e., correctly detected objects. Since precise bounding box localization is less critical in our context and manually generated annotations often exhibit positional variability, we additionally consider mAP at an IoU threshold of 0.25.

To unify multiple performance indicators into a single evaluation criterion, we build upon the *model fitness* concept introduced in [4, 69]. While [4] defines model fitness as an equally weighted mAP objective used for training optimization, we extend this concept by integrating FNR and FDR. Furthermore, we adapt the weighting scheme to our application context, defining the contribution of each component according to the aforementioned prioritization. Consequently, we formulate the model fitness as follows:

$$\begin{aligned} \text{fitness} = \; & 0.45 \times (1 - \mathrm{FNR}) \\ + \; & 0.35 \times (1 - \mathrm{FDR}) \\ + \; & 0.10 \times \mathrm{mAP}_{0.25} \\ + \; & 0.10 \times \mathrm{mAP}_{0.5} \; . \end{aligned} \tag{8}$$

Empirical evaluations show that variations in the absolute weight values have negligible impact, as long as their relative prioritization is maintained (cf. Table III, Supp. Mat.).

4.5 Experimental procedure

To assess the effectiveness of the proposed enhancements to YOLO-FEDER FusionNet, we conduct four structured experiments, each addressing a distinct focus area: training data optimization, feature integration, architectural modifications, and false positive mitigation. In addition, a final experiment benchmarks YOLO-FEDER FusionNet against state-of-the-art YOLO-based drone detection models and evaluates its cross-dataset generalization capabilities. To ensure comparability, a consistent training configuration is applied across all DL models (see Sec. 4.3). Evaluation is performed on the real-world datasets R1 and R2, following the evaluation protocol outlined in Sec. 4.4. The methodological approach for each experiment is presented in the following sections.

4.5.1 Exp. 1 – Training data optimization

To evaluate the impact of an optimized training dataset on the detection performance of YOLO-FEDER FusionNet, we build upon our previous findings presented in [64]. Specifically, we adopt a training dataset that combines a large-scale, photo-realistic synthetic dataset (SynDroneVision, see Sec. 4.1.3) with a small share of real-world data from the publicly available DUT Anti-UAV dataset (see Sec. 4.1.2). The resulting model is compared against the baseline YOLO-FEDER FusionNet, originally trained on the simpler synthetic dataset S1 (cf. Table 3), as introduced in [21]. For further comparison, we also include a generic YOLOv5 architecture trained using the same hybrid dataset configuration.

4.5.2 Exp. 2 – Impact analysis of intermediate FEDER features

By leveraging only FEDER's final output O_S (cf. Fig. 3), the original YOLO-FEDER FusionNet architecture overlooks potentially valuable intermediate features generated by internal components of the FEDER branch (e.g., the DWD module or the SED, cf. Fig. 3). To evaluate the contribution of intermediate FEDER feature representations, we extract feature maps from selected layers of the DWD module and the SED. These modules are designed to generate semantically rich hierarchical features, comparable to those obtained by the YOLO backbone. More specifically, we consider the feature maps O_S, F_i^D for $i \in \{2, \ldots, 4\}$ (from the DWD module), and F_j^S for $j \in \{1, \ldots, 4\}$ (from the SED) across six distinct fusion scenarios.

Feature fusion is performed at spatially aligned stages within the neck of YOLO-FEDER FusionNet, leveraging the hierarchical structure of FEDER features. By ensuring spatial compatibility, this alignment eliminates the need for resizing operations. This mitigates interpolation artifacts and preserves feature integrity. The integration process is facilitated by four attention-based fusion modules (AFM 1-4, Fig. 2), each implementing a two-stage mechanism consisting of feature concatenation followed by channel-wise attention. The adopted fusion strategy remains consistent with the approach originally proposed in YOLO-FEDER FusionNet [21]. An overview of the distinct fusion

configurations and their associated FEDER features is provided in Table 5.

4.5.3 Exp. 3 – Evaluation of YOLO- based backbone variations

YOLO-FEDER FusionNet's initial iteration employs YOLOv5l as backbone for generic, multi-scale feature extraction [21]. While this configuration has proven effective, more recent YOLO iterations (e.g., YOLOv8) have demonstrated enhanced detection accuracy on established benchmark datasets. To evaluate the architectural adaptability of YOLO-FEDER FusionNet and explore the potential benefits of incorporating more advanced feature extractors, we systematically replace the original YOLOv5l backbone with successor YOLO variants. To ensure methodological consistency, we select YOLO backbone architectures with trainable parameter counts closely aligned with that of YOLOv5l (cf. Table 1). Specifically, we include the backbone components of YOLOv8 (m and l), YOLOv9 (c and e), and YOLOv11 (l and x). The overall network architecture (cf. Fig. 2), the fusion strategy – specifically Config 4 (cf. Table 5) – as well as the training and evaluation conditions remain unchanged.

4.5.4 Exp. 4 – Detection error assessment

Building upon the trained models and insights from Exp. 3 (Sec. 4.5.3), this experiment provides a systematic evaluation of detection errors – focusing specifically on false positives and false negatives across distinct camera field-of-views (FOVs). The analysis combines quantitative statistical indicators – specifically, distribution patterns and average dimensions – with qualitative visual inspection to provide a comprehensive understanding of the network's detection behavior. Particular emphasis is placed on the architectural design of YOLO-FEDER FusionNet, characterized by a fixed input resolution of 640×640 pixels, a hierarchical downsampling technique, and a multi-scale detection head (cf. Sec. 3.4). The interplay of these architectural components introduces scale sensitivity, which can adversely affect the detection of small objects – particularly in wide FOV configurations. To further quantify the impact of this architectural constraint, we perform an additional analysis to investigate how the exclusion of small-scale objects – both from ground-truth annotations and model predictions – affects the overall detection performance. Following the categorization in [70], we adopt a minimum object size threshold of 16×16 pixels with respect to the original image resolution, representing extra-small objects. Instances below this threshold are excluded from evaluation.

Table 5 Overview of feature fusion configurations. Each configuration specifies the FEDER features integrated at the attention fusion modules (AFM 1-4, cf. Fig. 2), using the final segmentation output (O_S), as well as the intermediate feature maps F^D (DWD) and F^S (SED)

		Fusion Modules			
		AFM 1	AFM 2	AFM 3	AFM 4
Feature Integration	Config 1	O_S	O_S	O_S	O_S
	Config 2	F_1^S	F_1^S	F_1^S	F_1^S
	Config 3	–	F_2^S	F_3^S	F_4^S
	Config 4	–	F_2^D	F_3^D	F_4^D
	Config 5	–	F_2^S, F_2^D	F_3^S, F_3^D	F_4^S, F_4^D
	Config 6	O_S	O_S, F_2^D	O_S, F_3^D	O_S, F_4^D

4.5.5 Exp. 5 – Comparison to generic detectors & cross-dataset generalization

To evaluate the contribution of COD-based feature representations, YOLO-FEDER FusionNet is systematically benchmarked against representative generic YOLO-based detection architectures. In particular, the comparison includes YOLO configurations commonly adopted for image-based drone detection, namely YOLOv5, YOLOv8, and YOLOv9 (cf. Sec. 1). The models' architectural scales are aligned with those considered in the context of Exp. 3 (cf. Sec. 4.5.3).

To ensure consistency and comparability, all generic YOLO models are trained under experimental conditions directly comparable to those of YOLO-FEDER FusionNet. Beyond identical hyperparameter settings and data augmentation strategies (Secs. 4.3.1-4.3.2), the same data composition – combining real-world samples from DUT Anti-UAV with synthetic images from SynDroneVision (train/val splits; cf. Table 3) – is adopted as training foundation. As standard YOLO models natively operate on rectangular input images and do not impose square-input constraints – unlike YOLO-FEDER FusionNet – training (and evaluation) is performed on uncropped data.

Performance evaluation is conducted on R1 and R2 and further extended to DUT Anti-UAV (test split) to investigate generalization beyond the custom acquisition domain. Within this evaluation framework, all generic YOLO-based models are compared against the best-performing YOLO-FEDER FusionNet configuration identified in Exp. 1-3.

5 Results

This section provides a comprehensive evaluation and discussion of Exps. 1-5 (cf. Secs. 4.5.1-4.5.5), including quantitative performance assessment, systematic component ablation, and and cross-dataset generalization analysis with comparisons to generic drone detection models.

5.1 Training data optimization

The comparative analysis in Table 6 highlights the clear benefit of an optimized training dataset that combines large-scale, photo-realistic synthetic data (SynDroneVision [64]) with limited real-world samples (DUT Anti-UAV [35]). Models trained on the optimized dataset consistently achieve superior performance compared to those trained exclusively on the synthetic dataset S1 across all evaluation metrics.

YOLO-FEDER FusionNet demonstrates the highest overall performance when trained on the combination of SynDroneVision and DUT Anti-UAV, particularly at a cropping resolution of 640×640. In this configuration, the model achieves mAP values of 0.879 (R1) and 0.946 (R2) at an IoU threshold of 0.25 (cf. mAP@0.25, Table 6), along with notably low FNRs of 0.197 and 0.146, respectively. Most configurations also exhibit a decline in FDRs. However, an exception is observed on Dataset R2 at a cropping resolution of 1080×1080, where YOLO-FEDER FusionNet yields a comparatively higher FDR.

In comparison to the generic YOLOv5l detection model trained under identical conditions, YOLO-FEDER FusionNet consistently yields improved performance across key performance indicators. While YOLOv5l benefits from the enhanced training data, YOLO-FEDER FusionNet matches or exceeds this performance. Simultaneously, YOLO-FEDER FusionNet, trained on the optimized dataset, offers substantially lower FNRs and FDRs – most notably on the more challenging dataset R2.

5.2 Impact of intermediate FEDER feature representations

A systematic evaluation of the integration of diverse FEDER feature representations (cf. Sec. 4.5.2) across multiple datasets and cropping resolutions reveals notable variations in performance (see Table 7). These differences appear to be primarily driven by the characteristics of the extracted features, the stage of their integration, and the associated computational complexity (cf. Tables 8 and V, Supp. Mat.).

Among all evaluated fusion configurations, Config 4 – which leverages intermediate DWD features F_i^D, $i \in \{2, \ldots, 4\}$ – demonstrates the most favorable performance across all evaluation metrics. At a cropping resolution of 1080×1080, it achieves the highest mAP values at both IoU thresholds (0.25 and 0.5) and obtains the highest overall fitness scores across both datasets. Despite a marginal drop in fitness at 640×640 compared to the best-performing configuration (e.g., by 0.044 on Dataset R2), it remains highly competitive – ranking second across most key metrics. With an average score of 0.882, Config 4 achieves the highest overall fitness across all dataset-resolution combinations, indicating a strong overall trade-off between detection precision and generalization capability. At the same time, it maintains one of the lowest overall neck-head complexities (cf. Table V, Supp. Mat.), despite higher cost at AFM 2-4 due to increased input dimensionality (cf. Table 8).

In comparison, Config 1 – which exclusively incorporates the final FEDER output O_S at each AFM block (cf. Table 5) – exhibits competitive performance, particularly at the lower cropping resolution of 640×640. At this resolution, it yields the highest scores across all evaluation metrics on Dataset R1. Furthermore, it ranks second in fitness when averaged across all datasets and cropping resolutions, with an overall score of 0.873. The integration of intermediate SED features F_j^S, $j \in \{1, \ldots, 4\}$ (Configs. 2 and 3), demonstrates strong detection performance on Dataset R2, with low FDRs (0.011 and 0.009) and competitive mAP values. However, both configurations exhibit notably higher FDRs on Dataset R1, with values reaching up to 0.221 (cf. Table 7). Hence, neither configuration surpasses the average fitness score achieved by the integration of DWD features (Config 4).

Table 6 Performance comparison between YOLOv5l and YOLO-FEDER FusionNet, trained on S1 and a combination of SynDroneVision (SDV) [64] and DUT Anti-UAV (DUT) [35]. Best results across are highlighted in **bold**, while the second-best are underlined

Model	Training Data		Img Size	Dataset R1				Dataset R2			
	S1	SDV+DUT		mAP ↑		FNR ↓	FDR ↓	mAP ↑		FNR ↓	FDR↓
				@0.25	@0.5			@0.25	@0.5		
YOLOv5l	✓	–	2040×1086	0.572	0.551	0.463	0.500	0.571	0.432	0.745	0.290
	–	✓		0.847	0.826	0.241	**0.025**	0.829	0.692	0.376	0.025
	✓	–	1080×1080	0.568	0.548	0.457	0.261	0.685	0.270	0.473	0.029
	–	✓		0.861	0.819	0.199	0.073	0.805	0.645	0.440	0.038
	✓	–	640×640	0.433	0.401	0.601	0.311	0.102	0.047	0.858	0.638
	–	✓		0.747	0.684	0.345	0.040	0.574	0.408	0.647	0.047
YOLO-FEDER	✓	–	1080×1080	0.708	0.636	0.449	0.066	0.816	0.423	0.335	**0.007**
	–	✓		0.848	0.832	0.205	0.166	**0.946**	**0.774**	**0.139**	0.011
	✓	–	640×640	0.729	0.669	0.372	0.114	0.685	0.270	0.473	0.029
	–	✓		**0.879**	**0.852**	**0.197**	0.045	**0.946**	0.762	0.146	0.018

Table 7 Evaluation of FEDER backbone features and fusion configurations in YOLO-FEDER FusionNet, using a YOLOv5l backbone trained on SynDroneVision (SDV) [64] and DUT Anti-UAV (DUT) [35]. Best results are in **bold**, second-best are underlined. Fitness scores are given as absolute values and relative differences (percentage points) w.r.t. the baseline (Config 1). Configurations follow Table 5: 1 – O_S; 2 – F_1^S; 3 – SED features F_i^S, $i \in \{2,3,4\}$; 4 – DWD features F_i^D, $i \in \{2,3,4\}$; 5 – F_i^S and F_i^D, $i \in \{2,3,4\}$; 6 – O_S and F_i^D, $i \in \{2,3,4\}$

Img Size	Feat. Int. Config	Dataset R1						Dataset R2					
		mAP ↑		FNR ↓	FDR ↓	fitness ↑		mAP ↑		FNR ↓	FDR ↓	fitness ↑	
		@0.25	@0.5					@0.25	@0.5				
1080×1080	1	0.848	0.832	**0.205**	0.166	0.818		0.946	0.774	0.139	0.011	0.906	
	2	0.837	0.819	0.210	0.221	0.794	(−2.4)	0.953	0.790	0.143	0.011	0.906	(+0.0)
	3	0.839	0.821	0.228	0.179	0.801	(−1.7)	0.946	0.788	0.148	**0.009**	0.904	(−0.2)
	4	**0.853**	**0.835**	0.220	**0.094**	**0.837**	(+1.9)	**0.962**	**0.828**	**0.092**	0.013	**0.933**	(+2.7)
	5	0.816	0.800	0.225	0.294	0.758	(−6.0)	0.937	0.755	0.155	0.012	0.895	(−1.1)
	6	0.834	0.819	0.224	0.219	0.788	(−3.0)	0.935	0.750	0.162	0.015	0.890	(−1.6)
640×640	1	**0.879**	**0.852**	**0.197**	**0.045**	**0.869**		0.946	0.762	0.146	0.018	0.899	
	2	0.858	0.832	0.213	0.112	0.834	(−3.5)	**0.953**	0.798	0.156	0.007	0.903	(+0.4)
	3	0.838	0.813	0.244	0.068	0.832	(−3.7)	0.948	0.788	0.154	0.009	0.901	(+0.5)
	4	0.853	0.830	0.229	0.055	0.846	(−2.3)	0.944	**0.804**	0.121	0.026	0.911	(+1.2)
	5	0.851	0.830	0.233	0.077	0.836	(−3.3)	0.921	0.759	**0.025**	**0.005**	**0.955**	(+5.6)
	6	0.853	0.832	0.220	0.089	0.838	(−3.1)	0.912	0.730	0.166	0.070	0.865	(−3.4)

Table 8 AFM-level comparison of FEDER feature fusion configurations in terms of input channels (Ch.) and trainable parameters (Params)

Feat. Int. Config	AFM 1		AFM 2		AFM 3		AFM 4	
	Ch.	Params	Ch.	Params	Ch.	Params	Ch.	Params
1	1025	1.05	513	0.264	513	0.264	1025	1.052
2	1025	1.05	513	0.264	513	0.264	1025	1.052
3	1024	_★	513	0.264	513	0.264	1025	1.052
4	1024	_★	608	0.370	608	0.370	1120	1.256
5	1024	_★	609	0.371	609	0.371	1121	1.258
6	1025	1.02	609	0.371	609	0.371	1121	1.258

★ Concatenation only (no channel attention due to absent FEDER features)

The joint integration of DWD and SED features (Config 5) does not yield significant performance improvements over the exclusive use of DWD-derived features. Despite achieving the lowest FNR and FDR on Dataset R2 at a cropping resolution of 640×640, these improvements do not translate into enhanced overall performance. The limited effectiveness of incorporating additional SED features is also reflected in the lower average fitness score of 0.861. A similar observation can be made for Config 6, which incorporates the final binary segmentation output O_S alongside DWD features. Among all fusion configurations, Config 6 demonstrates the weakest overall performance, with an average fitness score of 0.845, while also exhibiting the highest computational complexity (cf. Table V, Supp. Mat.).

Overall, these findings indicate that leveraging intermediate DWD features (according to Config 4) offers the most substantial contribution to enhancing the performance of YOLO-FEDER FusionNet, especially under complex environmental conditions. Accordingly, this configuration is adopted as the default feature fusion strategy for all subsequent experiments (see Secs. 5.3 and 5.4).

5.3 Effectiveness of YOLO-based backbone variations

Among all YOLO backbone configurations evaluated within the YOLO-FEDER FusionNet architecture, the incorporation of YOLOv8l demonstrates the highest and most consistent detection performance across all experimental conditions (cf. Table 9). The YOLOv8l-enhanced YOLO-FEDER FusionNet achieves consistently high mAP values at an IoU threshold of 0.25 across all cropping resolutions and datasets. It also leads in mAP at an IoU threshold of 0.5 on Dataset R1. In addition, it exhibits strong overall fitness – with an average score of 0.894 – while maintaining low FNRs and FDRs. Specifically, it ranks among the top two configurations in terms FNR, with values ranging from 0.082 (R2, 640×640) to 0.192 (R1, 1080×1080). Importantly, these gains come without a measurable increase in inference time relative to the baseline (cf. Table 10).

Despite the clear performance advantages provided by the YOLOv8l backbone, the original YOLO-FEDER FusionNet configuration – featuring YOLOv5l – remains a strong and reliable baseline. It achieves the second-highest

Table 9 Performance evaluation of different YOLO backbones within the YOLO-FEDER FusionNet framework. The evaluated architectures follow the neck and head design depicted in Fig. 2, leveraging FEDER features F_i^D, $i \in \{2, \ldots, 4\}$ according to feature fusion configuration 4 (cf. Table 5). Best results are in **bold**, second-best are underlined. Fitness scores are given as absolute values and relative differences (percentage points) to the baseline (YOLOv5l backbone)

Img Size	YOLO Bckbn	Dataset R1						Dataset R2					
		mAP ↑		FNR ↓	FDR ↓	fitness ↑		mAP ↑		FNR ↓	FDR ↓	fitness ↑	
		@0.25	@0.5					@0.25	@0.5				
1080×1080	v5l	0.853	0.835	0.220	0.094	0.837		0.962	**0.828**	**0.092**	0.013	**0.933**	
	v8m	0.838	0.824	0.208	0.258	0.783	(−5.4)	0.959	0.817	0.112	**0.008**	0.924	(−0.9)
	v8l	**0.872**	**0.852**	**0.192**	0.049	**0.869**	(+3.2)	**0.967**	0.810	0.107	0.012	0.925	(−0.8)
	v9c	0.815	0.795	0.235	0.184	0.791	(−4.6)	0.953	0.802	**0.092**	0.020	0.927	(−0.6)
	v9e	0.852	0.838	0.215	0.086	0.842	(+0.5)	0.940	0.800	0.109	0.024	0.917	(−1.6)
	v11l	0.828	0.808	0.293	0.029	0.822	(−1.5)	0.896	0.760	0.138	0.163	0.847	(−8.6)
	v11x	0.803	0.777	0.333	**0.013**	0.804	(−3.3)	0.938	0.784	0.110	0.071	0.898	(−3.5)
640×640	v5l	0.853	0.830	0.229	0.055	0.846		0.944	0.804	0.121	0.026	0.911	
	v8m	0.836	0.813	0.211	0.173	0.809	(−3.7)	0.949	0.710	0.113	0.039	0.901	(−1.0)
	v8l	**0.876**	**0.848**	0.188	0.116	**0.847**	(+0.1)	**0.967**	0.798	**0.082**	0.019	**0.933**	(+2.2)
	v9c	0.830	0.806	0.242	0.095	0.822	(−2.4)	0.959	**0.806**	0.111	**0.007**	0.924	(+1.3)
	v9e	0.863	0.842	**0.179**	0.173	0.829	(−1.7)	0.943	0.770	0.105	0.040	0.910	(−0.1)
	v11l	0.847	0.822	0.250	**0.040**	0.840	(−0.6)	0.883	0.723	0.166	0.097	0.852	(−5.9)
	v11x	0.819	0.794	0.268	0.064	0.818	(−2.8)	0.935	0.754	0.102	0.078	0.896	(−1.5)

Table 10 Inference time comparison between YOLO-FEDER FusionNet and standard YOLO detectors across different YOLO variants. For each variant, YOLO-FEDER FusionNet is instantiated using the corresponding YOLO backbone, while YOLO denotes the standard end-to-end detection architecture. Values indicate average per-image inference time in milliseconds on an NVIDIA A100 GPU

Variant	YOLO-FEDER FusionNet★ (ms)	YOLO (ms)
v5l	12.5	2.6
v8m	11.7	1.9
v8l	12.4	2.9
v9c	12.4	3.1
v9e	16.1	7.0
v11l	12.0	2.7
v11x	12.9	4.5

★ DWD-based FEDER feature fusion (Config. 4)

average fitness score (0.882) and continues to perform competitively across other performance indicators. Nevertheless, it exhibits slightly elevated FNRs and FDRs than the YOLOv8l-based variant across the majority of evaluation scenarios. Additionally, it entails a higher model complexity, with a parameter count of 26.6M (cf. Table 1).

Similarly, the YOLOv9e-based configuration of YOLO-FEDER FusionNet achieves competitive results, particularly in terms of mAP and FNR, with the lowest observed FNR on Dataset R1. However, the strong localized performance fails to generalize into comprehensive performance improvements across all indicators and datasets (average fitness score: 0.875). Furthermore, the relatively high model complexity of the YOLOv9e backbone (30.2M parameters, 117.2B GFLOPs) adversely affects the computational efficiency of YOLO-FEDER FusionNet – especially in comparison to more balanced alternatives such as YOLOv8l. This effect is also reflected in the increased inference time (16.1 ms, cf. Table 10).

In contrast, the YOLOv8m-based configuration of YOLO-FEDER FusionNet constitutes the most computationally efficient variant, characterized by the lowest parameter count (11.8M), reduced computational cost (38.8B GFLOPs), and the shortest inference time (11.7 ms). However, these efficiency gains come at the cost of reduced detection performance, particularly in comparison to larger-backbone configurations. This reduction in performance is most pronounced in terms of FDR, reaching values of 0.173 and 0.258, respectively (cf. Dataset R1, Table 9). The lower average fitness score of 0.855 further reflects this underperformance, ranking it among the bottom three configurations. A comparable trend is observed for YOLO-FEDER FusionNet configured with a YOLOv9c backbone (cf. Table 9).

The integration of YOLOv11l and YOLOv11x backbones into YOLO-FEDER FusionNet yields the weakest performance across most evaluation metrics, with average fitness scores of 0.840 and 0.854, respectively. Despite comparable inference times (12.0 ms and 12.9 ms, cf. Table 10), the lack of performance gains limits their suitability within YOLO-FEDER FusionNet.

5.4 Detection error assessment

As previously noted (Sec. 5.3), YOLO-FEDER FusionNet configurations with YOLOv8m and YOLOv9c backbones exhibit elevated FDRs, with the highest rates observed on Dataset R1 (cf. Table 9). A visual analysis of the FPs,

Fig. 4 Spatial distribution of false positives generated by YOLO-FEDER FusionNet (YOLOv8m backbone + DWD-based FEDER features). The distribution is visualized across both camera FOVs of Dataset R1, using cropped regions of the full scenes. Zoomed-in regions highlight areas with high concentrations of recurring false positives across the majority of test samples. Color intensity encodes detection frequency, ranging from low (blue) to high (red)

conducted separately for each camera FOV, reveals systematic clustering of FPs within specific image regions (cf. Fig. 4). These regions primarily align with image areas affected by visual artifacts, including reflections, irregular texture patterns, or fine structural background details, which can create drone-like signatures.

A pixel-level semantic attribution of FP regions, mapping pixels associated with incorrect predictions to their underlying background categories, quantitatively supports this observation. In one camera FOV (cf. Fig. 4, right), FPs are predominantly associated with spatially dominant, visually complex semantic classes – namely buildings and vegetation – which jointly account for ∼90% of FP pixels. In the other camera FOV (cf. Fig. 4, left), FPs are more concentrated, with window regions contributing ∼75–80% of all FPs, indicating a pronounced sensitivity to reflective surfaces. Minor classes (e.g., localized structural primitives such as streetlights) contribute negligibly in terms of overall FP share (<5%), yet can exhibit elevated local FP densities, suggesting a disproportionate sensitivity to fine-grained structures.

Complementing this semantic analysis, the geometric characteristics of FPs provide additional insight. Further analysis of the bounding box dimensions indicates that FPs primarily correspond to small-scale artifacts, characterized by reduced width and height (see Fig. 5). For instance, YOLO-FEDER FusionNet configured with a YOLOv8m backbone produces FP bounding boxes with average dimensions of 21.2 pixels in width and 10.97 pixels in height. In contrast, TP detections exhibit mean width and height values of 44.28 and 21.51 pixels. A similar pattern is observed for the YOLOv9c-based backbone configuration.

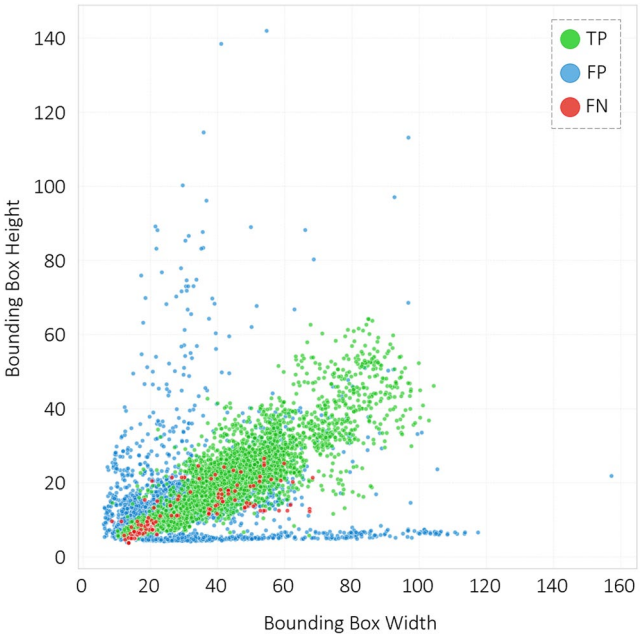

Fig. 5 Bounding box width vs. height for YOLO-FEDER FusionNet (YOLOv8m backbone) on R1 (1080×1080). FPs seem to be well-separable from TPs based on their smaller spatial dimensions. FNs overlap substantially with TPs, indicating that FNs arise from factors beyond object scale

Motivated by these observations, we introduce a size-based filtering strategy to mitigate erroneous detections, particularly those caused by small-scale artifacts. Applying a minimum object size threshold during post-processing – on both predictions and ground truth – leads to substantial improvements in error reduction. For instance, the FDR of the YOLOv8m-based YOLO-FEDER FusionNet drops from 0.258 to 0.067 on Dataset R1 (1080×1080 resolution;

Table 11 Performance evaluation of YOLO-FEDER FusionNet configurations (distinguished by backbone variations), considering only objects exceeding 16 × 16 pixels in size. The network architecture follows the design shown in Fig. 2, integrating FEDER features according to Config 4. Best results are highlighted in **bold**, second-best are underlined

Img Size	YOLO Bckbn	Dataset R1					Dataset R2				
		mAP ↑		FNR ↓	FDR ↓	fitness ↑	mAP ↑		FNR ↓	FDR ↓	fitness↑
		@0.25	@0.5				@0.25	@0.5			
1080×1080	v5l	0.881	0.877	0.223	0.016	0.870	0.927	0.798	0.139	0.010	0.906
	v8m	0.896	0.889	0.184	0.067	0.872	0.940	0.825	0.118	0.008	0.921
	v8l	**0.914**	**0.905**	**0.159**	0.017	**0.904**	0.942	0.811	0.111	0.012	0.921
	v9c	0.875	0.868	0.235	0.018	0.862	**0.946**	**0.839**	**0.103**	**0.009**	**0.929**
	v9e	0.890	0.887	0.186	0.075	0.868	0.937	0.838	0.115	0.024	0.917
	v11l	0.876	0.864	0.220	0.028	0.865	0.890	0.786	0.138	0.163	0.849
	v11x	0.852	0.842	0.272	**0.011**	0.843	0.929	0.821	0.117	0.070	0.898
640×640	v5l	0.882	0.876	0.227	0.009	0.870	0.911	0.770	0.111	0.012	0.914
	v8m	**0.936**	0.761	**0.115**	0.028	**0.908**	0.936	0.761	0.115	0.029	0.908
	v8l	0.896	0.884	0.202	0.009	0.884	**0.955**	0.822	**0.085**	0.009	**0.936**
	v9c	0.871	0.863	0.252	**0.005**	0.858	0.942	**0.834**	0.113	**0.007**	0.924
	v9e	0.911	**0.902**	0.164	0.028	0.898	0.941	0.810	0.106	0.039	0.914
	v11l	0.893	0.883	0.196	0.014	0.885	0.874	0.760	0.165	0.090	0.858
	v11x	0.880	0.870	0.227	0.015	0.868	0.936	0.809	0.103	0.061	0.907

Table 12 Comparison of YOLO-FEDER FusionNet (YOLOv8l backbone + DWD-based FEDER features) against generic YOLO-based detection models on the custom datasets R1 and R2 [63], where R2 is explicitly designed to represent almost exclusively camouflage-induced detection scenarios. **Bold**, underline, and *italic* denote the best, second-best, and third-best results per metric

Model	Crop Size	Dataset R1					Dataset R2				
		mAP ↑		FNR ↓	FDR ↓	fitness ↑	mAP ↑		FNR ↓	FDR ↓	fitness↑
		@0.25	@0.5				@0.25	@0.5			
YOLOv5l	–	0.848	0.827	0.241	0.038	0.846	0.864	0.704	0.345	0.014	0.797
YOLOv8m	–	**0.893**	**0.872**	0.241	**0.000**	*0.868*	0.905	0.734	0.330	**0.005**	0.814
YOLOv8l	–	0.868	0.847	*0.215*	0.008	**0.872**	*0.936*	0.688	*0.166*	0.006	*0.886*
YOLOv9c	–	0.868	0.847	0.239	*0.003*	0.863	0.928	0.695	0.406	0.009	0.776
YOLOv9e	–	0.866	0.847	0.233	0.001	0.866	0.866	0.616	0.351	**0.005**	0.789
YOLO-FEDER (v8l)	1080×1080	*0.872*	0.852	0.192	0.049	0.869	**0.967**	0.810	0.107	0.012	0.925
	640×640	0.876	*0.848*	**0.188**	0.116	0.847	0.955	**0.822**	**0.085**	*0.009*	**0.936**

cf. Tables 9 and 11). Similarly, the FDR of the YOLOv9c-based variant decreases significantly from 0.184 to 0.018. Excluding small-scale detections and corresponding ground-truth instances also improves detection performance in terms of mAP across all configurations. For instance, the YOLOv8l-based YOLO-FEDER FusionNet achieves mAP values of 0.914 (R1) and 0.942 (R2) at a cropping resolution of 1080×1080, while simultaneously reducing FNRs to 0.159 and 0.111, respectively. Furthermore, fitness scores improve substantially, reaching values up to 0.936 (R2, 640×640), surpassing previous bests (cf. Exp. 3, Table 9). Despite consistent improvements across all backbone configurations, the YOLOv8l-based YOLO-FEDER FusionNet maintains the highest average fitness score of 0.912.

5.5 Comparison to generic detectors & cross-dataset generalization

Under cross-domain evaluation, YOLO-FEDER FusionNet (YOLOv8l backbone + DWD-based FEDER features) demonstrates systematic performance improvements over conventional generic YOLO-based detection models across the majority of evaluation settings, while remaining competitive elsewhere (cf. Tables 12 and 13). The most pronounced gains are observed in the reduction of FNRs – a fundamental requirement for safety-critical applications. Even under the more heterogeneous visual conditions of dataset R1, which include both camouflage-prone backgrounds (e.g., vegetation) and visually favorable regions (e.g., sky and homogeneous building facades), YOLO-FEDER FusionNet achieves competitive values for mAP@0.25, mAP@0.5, and fitness (Table 12). Simultaneously, it maintains a considerably lower FNR than all YOLO variants.

Table 13 Comparison of YOLO-FEDER FusionNet (YOLOv8l backbone + DWD-based FEDER features) against generic YOLO-based detection models on the publicly available DUT Anti-UAV dataset [35], which targets general-purpose drone detection in urban environments. **Bold** and underline denote the best and second-best results per metric

Model	Crop Size	mAP ↑		FNR ↓	FDR ↓	fitness↑
		@0.25	@0.5			
YOLOv5l	–	0.957	0.937	0.102	0.035	0.933
YOLOv8m	–	0.960	0.938	0.117	0.025	0.928
YOLOv8l	–	0.963	0.944	0.105	0.015	0.938
YOLOv9c	–	0.961	0.941	0.107	0.022	0.934
YOLOv9e	–	0.969	0.955	0.095	**0.009**	0.947
YOLO-FEDER (v8l)	dyn.	**0.979**	**0.967**	**0.045**	0.032	**0.963**

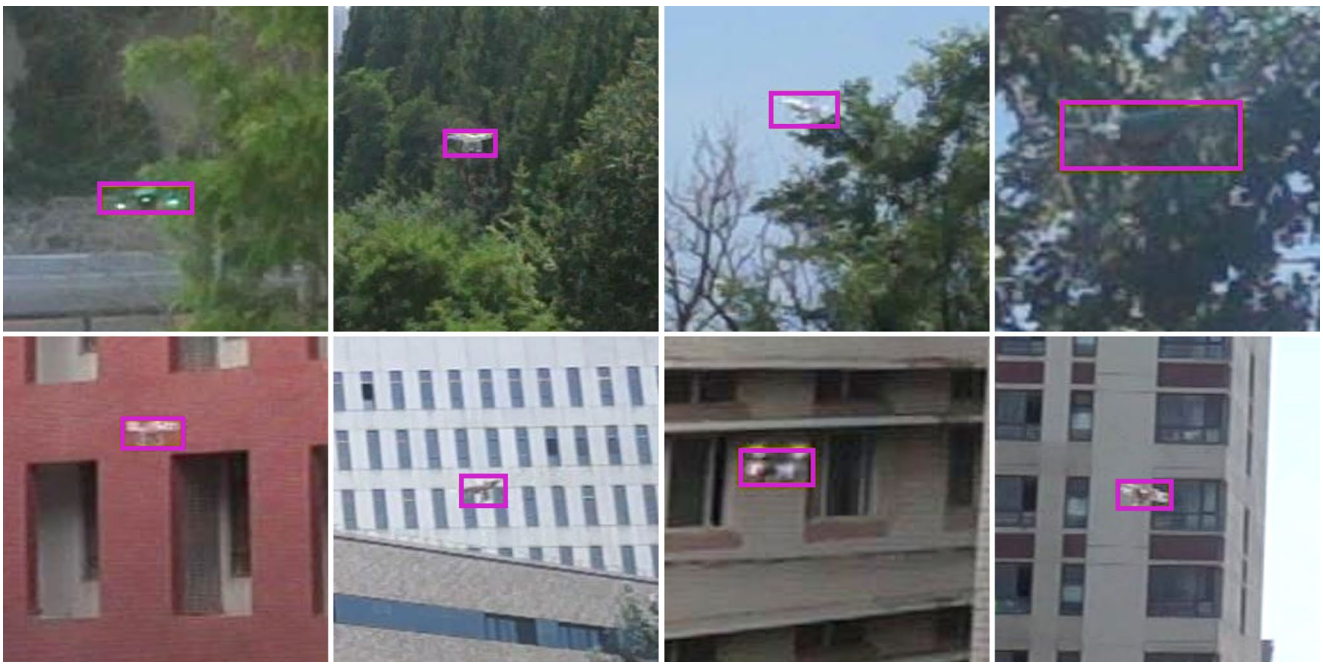

Fig. 6 Representative DUT Anti-UAV samples illustrating scenarios in which YOLOv9e (the strongest generic baseline on DUT Anti-UAV) fails to detect the drone, while YOLO-FEDER FusionNet (YOLOv8l backbone + DWD-based FEDER features) successfully localizes the target (magenta bounding boxes). Images are shown as cropped regions of the original frames, with slight brightness enhancement to improve small drone target visibility

However, the advantages of integrating COD-based cues alongside generic YOLO-derived feature representations become most evident on the camouflage-dominated dataset R2. Under these conditions, conventional YOLO detectors exhibit a substantial degradation in detection performance, reflected in markedly reduced mAP values (cf. mAP@0.5, Table 12) and sharply increased FNRs. Relative to the strongest generic baseline (YOLOv8l), YOLO-FEDER FusionNet increases mAP values by up to 0.134 and reduces the FNR by a factor of two (from 0.166 to 0.085). Overall, YOLO-FEDER FusionNet achieves the highest mAP scores and the best overall fitness among all evaluated models. The primary trade-off is a slight increase in FDR.

The promising performance of YOLO-FEDER FusionNet also extends to the public DUT Anti-UAV dataset, even though it represents more general drone detection scenarios without deliberate camouflage. Despite a reduced performance gap to generic YOLO models, YOLO-FEDER FusionNet continues to achieve the highest mAP values, the best overall fitness, and the lowest FNR. Figure 6 highlights the improved detection capability of YOLO-FEDER FusionNet compared to the leading generic baseline on DUT Anti-UAV. (Additional visual examples are provided in the Supp. Mat..)

Overall, YOLO-FEDER FusionNet seems to maintain a stable accuracy-robustness trade-off across all evaluated datasets, while the strongest generic YOLO baseline varies with the underlying data domain (YOLOv8m on R1, YOLOv8l on R2, YOLOv9e on DUT Anti-UAV).

5.6 Discussion

The results presented in Secs. 5.1-5.5 demonstrate that systematic optimization of both training data and network architecture substantially improves detection accuracy (cf. Table IV, Supp. Mat.). The combination of large-scale synthetic data with a limited amount of real-world samples (i.e., SynDroneVision [64] and DUT Anti-UAV [35]) significantly enhances the performance of YOLO-FEDER FusionNet, underscoring the importance of dataset diversity and realism (Sec. 5.1). These findings are consistent with prior work [64].

Systematic evaluation of FEDER feature representations shows that integrating intermediate DWD features yields the most consistent performance gains under our evaluation setup (Sec. 5.2). This highlights the importance of leveraging deeper, multi-scale feature information rather than relying solely on final outputs or auxiliary segmentation cues – as previously done in [21]. It also indicates that intermediate DWD features seem to offer more robust discriminative cues for detecting camouflaged drones compared to later-stage feature representations.

When considering backbone variations, the YOLOv8l-enhanced YOLO-FEDER FusionNet consistently achieves the most favorable trade-off between detection accuracy and computational efficiency (Sec. 5.3). While larger backbones such as YOLOv9e offer competitive accuracy, their increased complexity can hinder deployment in resource-constrained environments. Conversely, lightweight variants like YOLOv8m and YOLOv9c incur higher FDRs, reflecting a trade-off between model efficiency and detection reliability. Despite supporting a variety of YOLO backbones, YOLO-FEDER FusionNet does not necessarily benefit from the superior benchmark performance of newer architectures (e.g., YOLOv8l vs. YOLOv11l) and from the scaling gains typically achieved with larger model variants (e.g., YOLOv8m vs. YOLOv8l) on large-scale benchmark datasets. This behavior may stem from the decoupled use of backbone components, which potentially limits the network's ability to fully exploit the representational and optimization advantages of YOLO's end-to-end architecture – advantages that are specifically tuned to maximize performance on benchmark data (e.g., COCO [71]). Furthermore, the present application domain explicitly focuses on drone detection, where camouflage effects pose significant challenges for conventional generic object detection models – in contrast to COCO [71], where the primary challenge lies in distinguishing a large variety of object classes across diverse visual contexts. Consequently, the additional representational capacity of larger backbones may offer only limited performance gains in this domain.

A detailed analysis of FPs reveals an increased sensitivity to fine-grained background structures, particularly within visually complex regions, such as vegetation and reflective surfaces (cf. Sec. 5.4). These small-scale artifacts constitute a primary source of detection errors, frequently inducing drone-like activations. This behavior likely arises from YOLO-FEDER FusionNet's multi-scale detection strategy, which predicts drone instances at three spatial resolutions – 20×20, 40×40, and 80×80 – for an input size of 640×640. Due to successive downsampling, the detection of objects smaller than 8×8 pixels becomes unreliable, while feature discriminability is reduced for regions below ∼16×16 pixels [72]. This seems to lead to erroneous activations triggered by background noise and visual artifacts. Incorporating a size-filtering post-processing step mitigates these effects by suppressing small-scale false activations, leading to reduced FDRs – particularly in lightweight backbone configurations – while also improving mAP and overall fitness.

These findings further emphasize the adverse impact of small-scale background structures and small drone instances on detection performance [6, 9, 10, 23, 24], reaffirming that detecting small drones remains a persistent challenge. Addressing these limitations may require targeted architectural adaptations, such as incorporating an additional detection scale (e.g., 160×160 [10]), increasing the input resolution, or leveraging feature-space super-resolution techniques during training [73] to enhance fine-grained spatial representations.

When benchmarked against generic YOLO-based drone detection models, YOLO-FEDER FusionNet demonstrates superior performance in camouflage-dominated scenes, achieving substantially lower FNRs and high mAP values (Sec. 5.5) – occasionally at the cost of a slight increase in FDR. Moreover, the strong performance on more general datasets (e.g., DUT Anti-UAV) indicates that YOLO-FEDER FusionNet is able to learn camouflage-aware feature representations without compromising generic detection performance.

Finally, it should be emphasized that the most favorable configuration of YOLO-FEDER FusionNet identified in this work reflect the adopted metric priorities. Alternative application scenarios may impose different priorities, which in turn may require a task-specific re-evaluation of the model configuration.

6 Conclusion

In this work, we presented an enhanced version of YOLO-FEDER FusionNet for robust drone detection in visually complex environments. By systematically optimizing training data, FEDER feature integration, and the YOLO

backbone architecture, we achieved notable improvements in detection accuracy and robustness. Our findings highlight the importance of combining large-scale, photo-realistic synthetic training data with real-world samples, leveraging intermediate multi-scale FEDER features, and aligning post-processing strategies with architectural constraints. For our target application, YOLO-FEDER FusionNet configured with a YOLOv8l backbone and intermediate DWD features demonstrated the most favorable performance, particularly in comparison to generic YOLO-based models.

Future work will focus on targeted architectural refinements, such as higher-resolution detection heads and enhanced multi-scale feature aggregation, to further improve performance in visually complex environments. Moreover, incorporating complementary sensing modalities (e.g., infrared) represents a promising step toward enabling reliable drone detection across day and night.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00138-026-01856-3.

Author contributions All authors contributed to the conceptualization and methodology of the study. T.R.L. was responsible for software development, validation, formal analysis, data curation, visualization, and project administration. T.R.L. and A.W. conducted the investigation. A.W. and T.K. supervised the work. T.K. provided the resources and acquired the funding for the study. T.R.L. prepared the initial draft of the manuscript. All authors reviewed the manuscript and approved the final version.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability All datasets used in this study are publicly available and are appropriately cited within the manuscript.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Chiper, F.-L., Martian, A., Vladeanu, C., et al.: Drone Detection and Defense Systems: Survey and a Software-Defined Radio-Based Solution. Sensors (2022). https://doi.org/10.3390/s22041453
2. Mohsan, S.A.H., Othman, N.Q.H., Li, Y.: Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends. Intell. Serv. Robot. **16**, 109–137 (2023). https://doi.org/10.1007/s11370-022-00452-4
3. Elsayed, M., Reda, M., Mashaly, A.S., et al.: Review on Real-Time Drone Detection Based on Visual Band Electro-Optical (EO) Sensor. In: 10th Int. Conf. Intell. Comput. Inf. Syst., 57–65 (2021). https://doi.org/10.1109/ICICIS52592.2021.9694151
4. Ultralytics: Ultralytics YOLO. URL: https://docs.ultralytics.com/de (accessed: 2026-05-15)
5. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection (2020). https://arxiv.org/abs/2004.10934
6. Lv, Y., Ai, Z., Chen, M., et al.: High-Resolution Drone Detection Based on Background Difference and SAG-YOLOv5s. Sensors (2022). https://doi.org/10.3390/s22155825
7. Hosen, M.I., Kahraman, S.: Deep Learning-Based Efficient Drone Detection and Tracking in Real-Time. In: 33rd Signal Process. Commun. Appl. Conf., 1–4 (2025). https://doi.org/10.1109/SIU66497.2025.11111880
8. Rouhi, A., Umare, H., Patal, S., et al.: Long-Range Drone Detection Dataset. In: IEEE Int. Conf. Consum. Electron., 1–6 (2024). https://doi.org/10.1109/ICCE59016.2024.10444135
9. Kim, J.-H., Kim, N., Won, C.S.: High-Speed Drone Detection Based On Yolo-V8. In: IEEE Int. Conf. Acoust. Speech Signal Process., 1–2 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095516
10. Chen, M., Zheng, Z., Sun, H., et al.: YOLOv8-MDS: A YOLOv8-Based Multi-Distance Scale Drone Detection Network. J. Phys: Conf. Ser. **2891**(15), 152008 (2024). https://doi.org/10.1088/1742-6596/2891/15/152008
11. Yu, W., Mo, K.: YOLO-GCOF: A Lightweight Low-Altitude Drone Detection Model. IEEE Access **13**, 53053–53064 (2025). https://doi.org/10.1109/ACCESS.2025.3553477
12. Akkaya, T., Şen, S.H., Kılçarslan, M.: Comparative Analysis of Yolov8 and Florence-2 for Multimodal Drone Detection Using Infrared and Visible Image Fusion. In: Innov. Intell. Syst. and Appl. Conf., 1–6 (2025). https://doi.org/10.1109/ASYU67174.2025.11208289
13. Meena, B.P., Manikandan, M.S.: Effectiveness of YOLO11-Based Lightweight Model for Drone Detection in Noisy Environments. In: 17th Int. Conf. Electron. Comput. Artif. Intell., 1–8 (2025). https://doi.org/10.1109/ECAI65401.2025.11095554
14. Zhou, S., Yang, L., Liu, H., et al.: A Lightweight Drone Detection Method Integrated into a Linear Attention Mechanism Based on Improved YOLOv11. Remote Sens. (2025). https://doi.org/10.3390/rs17040705
15. Riftiarrasyid, M.F., Gaol, F.L., Soeparno, H., et al.: Suitability of Latest Version of YOLOv11 in Drone Development Studies. In: 7th Int. Semin. Res. Inf. Technol. Intell. Syst., 504–509 (2024). https://doi.org/10.1109/ISRITI64779.2024.10963542
16. Zhao, Y., Lv, W., Xu, S., et al.: DETRs Beat YOLOs on Real-time Object Detection. In: IEEE/CVF Conf. Comput. Vis. Pattern

Recognit., 16965–16974 (2024). https://doi.org/10.1109/CVPR52733.2024.01605
17. Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., et al.: Real-Time and Accurate Drone Detection in a Video with a Static Background. Sensors (2020). https://doi.org/10.3390/s20143856
18. Dieter, T.R., Weinmann, A., Jäger, S., et al.: Quantifying the Simulation-Reality Gap for Deep Learning-Based Drone Detection. Electronics (2023). https://doi.org/10.3390/electronics12102197
19. Liu, Z., An, P., Yang, Y., et al.: Vision-Based Drone Detection in Complex Environments: A Survey. Drones (2024). https://doi.org/10.3390/drones8110643
20. Fan, D.-P., Ji, G.-P., Sun, G., et al.: Camouflaged Object Detection. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2774–2784 (2020). https://doi.org/10.1109/CVPR42600.2020.00285
21. Lenhard, T.R., Weinmann, A., Jäger, S., et al.: YOLO-FEDER FusionNet: A Novel Deep Learning Architecture for Drone Detection. In: IEEE Int. Conf. Image Process., 2299–2305 (2024). https://doi.org/10.1109/ICIP51287.2024.10647355
22. He, C., Li, K., Zhang, Y., et al.: Camouflaged Object Detection with Feature Decomposition and Edge Reconstruction. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 22046–22055 (2023). https://doi.org/10.1109/CVPR52729.2023.02111
23. Liu, H., Fan, K., Ouyang, Q., et al.: Real-Time Small Drones Detection Based on Pruned YOLOv4. Sensors (2021). https://doi.org/10.3390/s21103374
24. Dong, P., Wang, C., Lu, Z., et al.: S-Feature Pyramid Network and Attention Model for Drone Detection. In: IEEE Int. Conf. Acoust. Speech Signal Process., 1–2 (2023). https://doi.org/10.1109/ICASSP49357.2023.10096226
25. Yang, L., Zhang, R.-Y., Li, L.: SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In: 38th Int. Conf. Mach. Learn. Proceedings of Machine Learning Research, 139, 11863–11874. (2021)
26. Han, K., Wang, Y., Tian, Q., et al.: GhostNet: More Features From Cheap Operations. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 1577–1586 (2020). https://doi.org/10.1109/CVPR.42600.2020.00165
27. Li, H., Li, J., Wei, H., et al.: Slim-neck by GSConv: A Lightweight-design for Real-time Detector Architectures. J. Real Time Image Process. **21**, 62 (2022). https://doi.org/10.1007/s11554-024-01436-6
28. Gong, W.: Lightweight Object Detection: A Study Based on YOLOv7 Integrated with ShuffleNetv2 and Vision Transformer (2024). https://arxiv.org/abs/2403.01736
29. Chen, Y., Zhang, C., Chen, B., et al.: Accurate Leukocyte Detection Based on Deformable-DETR and Multi-level Feature Fusion for Aiding Diagnosis of Blood Diseases. Comput. Biol. Med. **170**, 107917 (2024). https://doi.org/10.1016/j.compbiomed.2024.107917
30. Chen, Y., Aggarwal, P., Choi, J., et al.: A Deep Learning Approach to Drone Monitoring. In: Asia–Pac. Signal Inf. Process. Assoc. Annu. Summit Conf., 686–691 (2017). https://doi.org/10.1109/APSIPA.2017.8282120
31. Svanström, F., Alonso-Fernandez, F., Englund, C.: Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities. Drones (2022). https://doi.org/10.3390/drones6110317
32. Lenhard, T.R., Weinmann, A., Snoussi, H.: Detector-Augmented SAMURAI for Long-Duration Drone Tracking. In: IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops, (2026)
33. Cheng, F., Liang, Z., Peng, G., et al.: An Anti-UAV Long-Term Tracking Method with Hybrid Attention Mechanism and Hierarchical Discriminator. Sensors (2022). https://doi.org/10.3390/s22103701
34. Wang, C., Shi, Z., Meng, L., et al.: Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism. Drones (2022). https://doi.org/10.3390/drones6060149
35. Zhao, J., Zhang, J., Li, D., et al.: Vision-Based Anti-UAV Detection and Tracking. IEEE Trans. Intell. Transp. Syst. **23**(12), 25323–25334 (2022). https://doi.org/10.1109/TITS.2022.3177627
36. Yang, Z., Yang, Y.: Decoupling Features in Hierarchical Propagation for Video Object Segmentation. In: NeurIPS (2022)
37. Zhu, J., Chen, Z., Hao, Z., et al.: Tracking Anything in High Quality (2023). https://arxiv.org/abs/2307.13974
38. Cheng, Y., Li, L., Xu, Y., et al.: Segment and Track Anything (2023). https://arxiv.org/abs/2305.06558
39. Yang, C.-Y., Huang, H.-W., Chai, W., et al.: SAMURAI: Adapting Segment Anything Model for Zero-Shot Visual Tracking with Motion-Aware Memory (2024). https://arxiv.org/abs/2411.11922
40. Jia, Q., Yao, S., Liu, Y., et al.: Segment, Magnify and Reiterate: Detecting Camouflaged Objects the Hard Way. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 4703–4712 (2022). https://doi.org/10.1109/CVPR52688.2022.00467
41. Yang, J., Zhong, B., Liang, Q.: Uncertainty-Guided Diffusion Model for Camouflaged Object Detection. IEEE Trans. Multimedia **27**, 4656–4669 (2025). https://doi.org/10.1109/TMM.2025.3535290
42. Sun, K., Chen, Z., Lin, X., et al.: Conditional Diffusion Models for Camouflaged and Salient Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. **47**(4), 2833–2848 (2025). https://doi.org/10.1109/TPAMI.2025.3527469
43. Zhou, Y., Li, Y., Fu, Y.: CamSAM2: Segment Anything Accurately in Camouflaged Videos. NeurIPS , (2025)
44. Yao, S., Sun, H., Xiang, T.-Z.: Hierarchical Graph Interaction Transformer With Dynamic Token Clustering for Camouflaged Object Detection. IEEE Trans. Image Process. **33**, 5936–5948 (2024). https://doi.org/10.1109/TIP.2024.3475219
45. Wang, Z., Deng, Y., Shen, C., et al.: Camouflaged Object Detection via Context and Texture-aware Hierarchical Interaction. Sci. Rep. **16**, 9328 (2026). https://doi.org/10.1038/s41598-025-32409-9
46. Zhang, S., Kong, D., Xing, Y.: Frequency-Guided Spatial Adaptation for Camouflaged Object Detection. IEEE Trans. Multimedia **27**, 72–83 (2025). https://doi.org/10.1109/TMM.2024.3521681
47. Barisic, A., Petric, F., Bogdan, S.: Sim2Air - Synthetic Aerial Dataset for UAV Monitoring. IEEE Robot. Autom. Lett. **7**(2), 3757–3764 (2022). https://doi.org/10.1109/LRA.2022.3147337
48. Marez, D., Borden, S., Nans, L.: UAV Detection with a Dataset Augmented by Domain Randomization. In: SPIE Geospat. Inf. X, 11398, 1139807 (2020). https://doi.org/10.1117/12.2558864
49. Symeonidis, C., Anastasiadis, C., Nikolaidis, N.: A UAV Video Data Generation Framework for Improved Robustness of UAV Detection Methods. In: IEEE 24th Int. Workshop Multimed. Signal Process., 1–5 (2022). https://doi.org/10.1109/MMSP55362.2022.9949360
50. Pham, H.X., Sarabakha, A., Odnoshyvkin, M., et al.: PencilNet: Zero-Shot Sim-to-Real Transfer Learning for Robust Gate Perception in Autonomous Drone Racing. IEEE Robot. Autom. Lett. **7**(4), 11847–11854 (2022). https://doi.org/10.1109/LRA.2022.3207545
51. Dieter, T.R., Weinmann, A., Brucherseifer, E.: Generating Synthetic Data for Deep Learning-Based Drone Detection. AIP Conf. Proc. **2939**(1), 030007 (2023). https://doi.org/10.1063/5.0180345
52. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement (2018). https://arxiv.org/abs/1804.02767
53. Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., et al.: CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 1571–1580 (2020). https://doi.org/10.1109/CVPRW50498.2020.00203
54. Wang, C.-Y., Yeh, I.-H., Mark Liao, H.-Y.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient

Information. In: Eur. Conf. Comput. Vis., 1–21 (2024). https://doi.org/10.1007/978-3-031-72751-1_1
55. Wang, C.-Y., Liao, H.-Y.M., Yeh, I.-H.: Designing Network Design Strategies Through Gradient Path Analysis. J. Inf. Sci, Eng (2023)
56. Gao, S., Cheng, M.-M., Zhao, K.: Res2Net: A New Multi-Scale Backbone Architecture. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 652–662 (2019). https://doi.org/10.1109/TPAMI.2019.2938758
57. Chen, L.-C., Papandreou, G., Kokkinos, I.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**, 834–848 (2016). https://doi.org/10.1109/TPAMI.2017.2699184
58. Stevens, M., Merilaita, S.: Animal Camouflage: Current Issues and New Perspectives. Philos. Trans. R. Soc. B Biol. Sci. 364(1516), 423–427 (2009)
59. Ha, W., Singh, C., Lanusse, F.: Adaptive Wavelet Distillation from Neural Networks Through Interpretations. In: NeurIPS, 20669–20682. (2021)
60. Weinan, E.: A Proposal on Machine Learning via Dynamical Systems. Commun. Math. Stat **5**, 1–11 (2017). https://doi.org/10.1007/s40304-017-0103-z
61. Haber, E., Ruthotto, L.: Stable Architectures for Deep Neural Networks. Inverse Probl. **34**(1), 014004 (2017). https://doi.org/10.1088/1361-6420/aa9a90
62. Woo, S., Park, J., Lee, J.-Y., et al.: CBAM: Convolutional Block Attention Module. In: Eur. Conf. Comput. Vis., 3–19 (2018). https://doi.org/10.1007/978-3-030-01234-2_1
63. Lenhard, T.R., Weinmann, A., Snoussi, H., et al.: Long-Duration Drone Tracking Dataset. Zenodo (2026). https://doi.org/10.5281/zenodo.17182190
64. Lenhard, T.R., Weinmann, A., Franke, K., et al.: SynDroneVision: A Synthetic Dataset for Image-Based Drone Detection. In: IEEE/CVF Winter Conf. Appl. Comput. Vis., 7637–7647 (2025). https://doi.org/10.1109/WACV61041.2025.00742
65. CVAT.ai Corporation: Open Data Annotation Platform. https://www.cvat.ai (accessed: 2026-04-15)
66. Epic Games: Unreal Engine. URL: https://www.unrealengine.com/en-US/ (accessed: 2026-04-15)
67. Codex Laboratories LLC: Welcome to Colosseum. https://github.com/CodexLabsLLC/ Colosseum/ (accessed: 2026-04-15)
68. Microsoft Research: Welcome to AirSim. https://microsoft.github.io/AirSim/ (accessed: 2026-04-15)
69. Silva, D., Jourdan, N., Gählert, N.: Long Range Object-Level Monocular Depth Estimation for UAVs. Image Anal , 325–340 (2023)
70. Hao, H., Peng, Y., Ye, Z., et al.: A High Performance Air-to-Air Unmanned Aerial Vehicle Target Detection Model. Drones (2025). https://doi.org/10.3390/drones9020154
71. Lin, T.-Y., Maire, M., Belongie, S.J., et al.: Microsoft COCO: Common Objects in Context. In: Eur. Conf. Comput. Vis., 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48
72. Lyu, Y., Liu, Z., Li, H., et al.: A Real-time and Lightweight Method for Tiny Airborne Object Detection. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 3016–3025 (2023). https://doi.org/10.1109/CVPRW59228.2023.00303
73. Wu, X., Hong, D., Chanussot, J.: UIU-Net: U-Net in U-Net for Infrared Small Object Detection. IEEE Trans. Image Process. **32**, 364–376 (2023). https://doi.org/10.1109/TIP.2022.3228497

Tamara R. Lenhard Tamara R. Lenhard received the M.Sc. degree in Applied Mathematics from Darmstadt University of Applied Sciences, Germany. She is currently a Research Associate with the German Aerospace Center (DLR), Institute for the Protection of Terrestrial Infrastructures, and is pursuing the Ph.D. degree at Darmstadt University of Applied Sciences in cooperation with DLR. Her research interests include computer vision, deep learning, synthetic data generation, and digital twin technologies. Her work focuses on the development of deep learning (DL) methods for the analysis of visual data, addressing applications such as drone detection and the semantic consistency between digital twins and reality. A further research focus is the generation of high-fidelity synthetic datasets using game-engine-based simulation to support the training of DL models.

Andreas Weinmann Andreas Weinmann received the Diplom degree in Mathematics, with a minor in Computer Science, from Technische Universität München in 2006 and the PhD degree from Technische Universität Graz in 2010. He worked as a researcher at the Technische Universität München and the Helmholtz Center Munich. From 2015 to 2025, he was a professor of Mathematics with a focus on Image Processing at Hochschule Darmstadt. Since 2025, he is a research professor for AI and Algorithms at Technische Hochschule Würzburg-Schweinfurt. His research interests lie at the intersection of mathematics, computer science, and engineering. In particular, he is interested in the development, application, and analysis of algorithms for signal and image processing, computer vision, and biomedical imaging. Further interests include the modeling and analysis of such algorithms. His work involves numerical optimization, regularization techniques for inverse problems, and machine learning based on deep neural networks.

Tobias Koch Tobias Koch is Head of the Department for Digital Twins of Infrastructures at the Institute for the Protection of terrestrial Infrastructures, German Aerospace Center (DLR). He holds a Ph.D. in engineering from the University of the Bundeswehr Munich. His research focuses on the automated generation of digital twins using artificial intelligence, scalable digital twin architectures, applications in crisis management, and human-machine interaction.