

Article

FCD-DETR: A Foreground-Aware and Context-Enhanced Detection Transformer for Pest Detection in Ultraviolet Light-Trap Images

Xiang Liu ^{1,†}, Binhong Zhou ^{1,†}, Qinshan Jiang ¹, Yan Cheng ², Yanxi Liu ¹ and Zhiyong Li ^{1,*} 

¹ College of Information Engineering, Sichuan Agricultural University, Ya'an 625014, China; 202406804@stu.sicau.edu.cn (X.L.); 202406649@stu.sicau.edu.cn (B.Z.); 202502247@stu.sicau.edu.cn (Q.J.); 202406802@stu.sicau.edu.cn (Y.L.)

² PipeChina Northwest Branch, Xi'an 710016, China; chengyan02@pipechina.com.cn

* Correspondence: lzy@sicau.edu.cn

[†] These authors contributed equally to this work.

Abstract

Ultraviolet (UV) insect traps enable continuous field pest monitoring, but their images contain severe scale variation, dense and overlapping targets, incomplete specimens, background stains, insect fragments, non-target insects, inter-class similarity, and long-tailed distributions, leading to missed detections, false positives, and class confusion. To address these challenges, we constructed a real-world UV insect-trap pest dataset comprising 12,426 high-resolution images and 38,003 annotated instances from 36 pest categories, and propose FCD-DETR, a foreground-aware and context-enhanced Detection Transformer based on RT-DETR-R18. The main distinction of FCD-DETR lies in a progressive enhancement design tailored to UV insect-trap images, where FADM first reduces foreground-background confusion by decoupling pest foreground cues from multi-scale semantic information, C2f_CFBBlock strengthens local-detail and contextual representations in the lightweight backbone, and DHSA further alleviates high-level semantic ambiguity through dynamic-range histogram self-attention in the Transformer encoder. On the proposed dataset, FCD-DETR achieved 64.42% *mAP@0.5* and 41.33% *mAP@50:95*, improving the baseline by 5.25 and 5.11 percentage points, respectively, and outperforming representative CNN- and Transformer-based detectors. Comprehensive experimental analyses confirm that FCD-DETR improves foreground discrimination and detection robustness in complex UV insect-trap scenarios.

Keywords: pest detection; UV insect-trap images; Detection Transformer; foreground-aware feature fusion; multi-scale feature fusion; context-enhanced modeling; smart agriculture



Academic Editor: Chengming Sun

Received: 2 June 2026

Revised: 4 July 2026

Accepted: 10 July 2026

Published: 12 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Agricultural pests threaten crop yield, product quality, and agricultural production security by directly damaging plant tissues and transmitting pathogens. Timely and accurate pest identification is therefore essential for precision pest management, green plant protection, and smart agricultural decision-making. However, conventional monitoring still relies heavily on manual surveys and expert identification, which are labor-intensive, subjective, time-consuming, and difficult to scale for continuous regional monitoring. With advances in automated trapping devices, embedded imaging systems, and deep learning,

image-based pest detection has become an important approach for intelligent agricultural monitoring [1,2].

In practical pest monitoring scenarios, ultraviolet (UV) insect trap systems exploit insect phototaxis to continuously attract field pests and collect images, providing a stable data source for deep learning-based pest detection. Unlike manually captured images, UV insect-trap images are usually collected in real-world field environments and often contain non-target insects, insect remains, trap grids, stains, and other background interference. Zhang et al. [3] proposed AgriPest-YOLO for light-trap agricultural pest detection, and Chen et al. [4] proposed DAMI-YOLOv8l to enhance multiscale feature representation in light-trap pest images. More recently, Tang et al. [5] proposed EP-YOLO for micro-pest detection in agricultural light-trap environments, further demonstrating the practical value of light-trap pest monitoring.

In recent years, deep learning-based object detection methods, especially YOLO-based single-stage detectors, have been widely applied to agricultural pest detection because of their high detection speed and end-to-end prediction capability [2]. Recent studies have improved pest detection performance through feature extraction enhancement, attention mechanisms, bottleneck optimization, multiscale representation, and lightweight design [6–9]. These studies show that structural optimization, attention modeling, and multiscale feature fusion are effective for agricultural pest detection.

However, multi-class pest detection using high-resolution UV insect-trap images still faces several coupled challenges. Chen et al. [10] reported that dense distribution, target similarity, and scale variation are key factors affecting multi-class dense pest detection and counting. In addition, category imbalance or long-tailed distribution can weaken the learning of low-frequency pest categories [11]. In UV insect-trap images, these challenges are further combined with complex background interference, making pest detection a comprehensive task involving foreground–background discrimination, multi-scale localization, fine-grained inter-class recognition, and long-tailed category learning. The main challenges are summarized in Table 1.

Table 1. Main challenges in UV insect-trap pest detection.

Challenge	Main Impact
Scale variation	Causes missed detections and localization errors.
Dense distribution and occlusion	Makes adjacent or incomplete pests difficult to separate.
Background interference	Increases background-induced false positives.
Inter-class similarity	Leads to category confusion among visually similar pests.
Long-tailed distribution	Weakens the detection of low-frequency pest categories.

Transformer-based detectors provide an alternative paradigm to CNN-based detectors by modeling object detection as an end-to-end set prediction problem. DETR [12] reduces the reliance on hand-designed detection components, Deformable DETR [13] improves multi-scale feature processing through deformable attention, and RT-DETR [14] further enhances real-time detection efficiency. These properties make RT-DETR a promising baseline for agricultural pest detection. Recent studies have also explored Transformer- or RT-DETR-based models for pest detection. Liu et al. [15] proposed RicePest-DETR for small, densely distributed, and multi-scale rice pest detection. Luo et al. [16] proposed CALDS-RTDETR, a robust RT-DETR-based detector for small forestry pests in complex environments. However, directly applying RT-DETR to UV insect-trap images remains insufficient because existing feature fusion and attention mechanisms may fail to clearly

distinguish valid pest foregrounds from stains, fragments, trap grids, non-target insects, and visually similar species, especially in dense and cluttered monitoring scenes.

To address these gaps, we propose FCD-DETR, a foreground-aware and context-enhanced Detection Transformer for pest detection in UV insect-trap images. Built on RT-DETR-R18, FCD-DETR improves the baseline from three aspects: foreground-aware multi-scale fusion, lightweight context-enhanced backbone feature extraction, and encoder-level semantic interaction. Specifically, FADM decouples foreground cues from multi-scale semantic information before the de-coder, C2f_CFBBlock enhances local-detail and contextual feature extraction in the backbone, and the DHSA-enhanced encoder strengthens high-level semantic discrimination in complex backgrounds. The main contributions of this paper are as follows:

- We constructed a pest detection dataset using UV insect traps tailored for real-world field pest monitoring scenarios. The dataset comprises 12,426 high-resolution images and 38,003 annotated pest instances, covering 36 pest categories. It exhibits characteristics such as varying target scales, dense distributions, background interference, non-target insects, inter-class similarity, and a long-tail distribution of categories.
- We propose the Foreground-Aware Decoupled Mixer (FADM) for foreground-aware multi-scale feature fusion. By decoupling foreground cues from multi-scale semantic information, this module enhances feature responses in effective pest regions while reducing interference from stains, non-target insects, and complex backgrounds on detection results.
- We construct a C2f_CFBBlock-based lightweight context-enhanced backbone. This architecture is based on C2f, with context-enhanced reconstruction of its internal bottleneck feature extraction units. While retaining the advantages of feature branching and fusion, it reduces the number of backbone parameters and improves the ability to capture local details and express contextual features in complex pest images.
- We designed a DHSA-enhanced Transformer encoder layer, integrating DHSA-based attention modeling into the RT-DETR encoder. While retaining residual connections, LayerNorm, and FFN, this approach enhances the contextual interaction capabilities of high-level semantic features, thereby improving the model's ability to distinguish between morphologically similar, occluded, and overlapping pest targets.
- Extensive experiments, including external validation on the public Pest24 dataset, demonstrate that FCD-DETR outperforms representative CNN- and Transformer-based detectors, while ablation studies verify the effectiveness of the proposed modules.

2. Materials and Methods

2.1. Dataset Construction

To support multi-class pest detection under real field monitoring conditions, we constructed a UV insect-trap pest dataset. The dataset construction workflow included image acquisition, manual annotation, quality control, and dataset partitioning. All images were collected from operational pest monitoring scenarios, thereby preserving typical visual challenges in light-trap imagery, including complex backgrounds, scale variation, dense target distributions, partial occlusion, non-target insects, and class imbalance.

2.1.1. Image Acquisition

The pest images were collected in Meishan City, Sichuan Province, China, and the monitored crop type was rice. The data collection period ranged from 15 April 2020 to 11 August 2024, covering approximately 4.3 years and spanning spring, summer, autumn, and winter. Image acquisition was performed using an RNCB-III ultraviolet insect trap system (Hunan RNXN Tech, Changsha, China), with each trap equipped with an LED light source

with a central wavelength of 360–400 nm. Images were automatically captured by the high-definition imaging unit integrated into the pest monitoring system, and the imaged specimens were field insects attracted and collected by the UV traps during monitoring.

The raw images had high spatial resolutions ranging from 4000×2550 to 5472×3648 pixels. These images retained fine-grained morphological details of pest targets, including wing shape, body structure, texture, and local edges, while also recording realistic environmental and weather-related interference factors, such as variations in illumination and humidity, rainfall-induced trap stains, dew or moisture on the trap plate, trap-plate grids, insect fragments, and non-target insects.

The image timestamps ranged from 00:00:03 to 23:59:48, covering a full diurnal cycle, with most images acquired between 20:00 and 02:59, consistent with the nocturnal activity of phototactic insects attracted by UV light traps. Figure 1 shows representative samples and enlarged regions of the 36 pest categories, illustrating variations in target scale, pose, color, morphology, occlusion, and background interference.

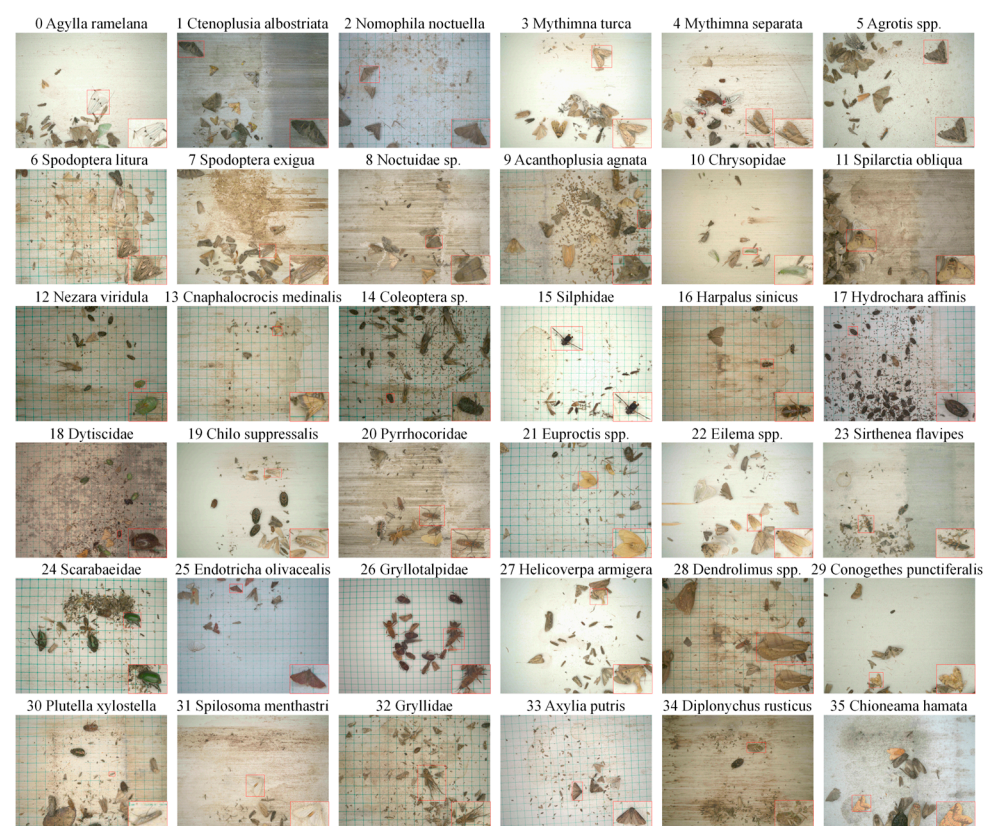


Figure 1. Representative samples from the UV insect-trap pest dataset and enlarged regions showing morphological details of the 36 pest categories.

2.1.2. Data Annotation and Quality Control

All images were manually annotated using LabelImg v1.8.1, and the annotation files were saved in YOLO format. During annotation, each identifiable target pest individual was labeled with a rectangular bounding box and assigned a corresponding class label, whereas interfering areas such as non-target insects, background stains, trap grids, and insect remains were not annotated if they did not belong to the 36 target pest categories defined in this study.

To improve annotation consistency and reliability, we adopted an independent review-based quality control procedure, in which a second annotator checked the annotation files after initial labeling, focusing on class-label accuracy, missed targets, and bounding-box

alignment. Samples with incorrect labels, missing annotations, or inaccurate bounding boxes were corrected. Pest individuals with slight occlusion or partial body damage were retained if their categories could be reliably identified. In contrast, severely blurred, heavily occluded, excessively damaged, or unidentifiable individuals were excluded. After annotation and quality control, the final dataset contained 38,003 annotated pest instances from 36 pest categories.

2.1.3. Dataset Split and Class Distribution

After annotation and quality control, the dataset was divided into training, validation, and test sets for model training, model selection, and final performance evaluation, respectively. The final dataset contained 12,426 images and 38,003 annotated pest instances, with detailed partition statistics provided in Table 2.

Table 2. Dataset partition statistics.

Split	Images	Instances
Train	9989	28,911
Validation	1225	4219
Test	1212	4873
Total	12,426	38,003

The dataset encompasses 36 pest categories, with the instance distribution illustrated in Figure 2. The frequency of occurrences differs significantly among categories. Class C24 comprises 8870 occurrences, while class C14 consists of merely 81 examples, signifying a marked long-tailed class distribution.

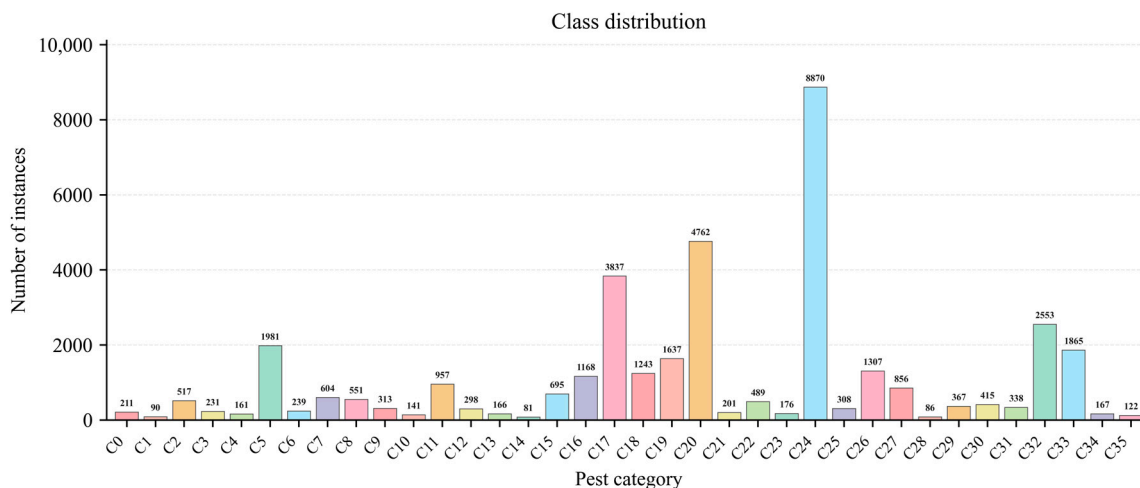


Figure 2. Distribution of category instances throughout the UV insect trap dataset.

This long-tailed distribution illustrates the inherent disparity in pest incidence frequencies observed during actual field monitoring circumstances. High-frequency categories offer plentiful training data, but low-frequency categories are more prone to inadequate feature learning, unstable detection, and class ambiguity. Consequently, the dataset facilitates both the comprehensive assessment of multi-class pest detection algorithms and the examination of model robustness in class-imbalanced scenarios.

2.1.4. Pest24 Dataset

To further evaluate the generalization ability of FCD-DETR, we additionally used the public Pest24 dataset as an external benchmark. Pest24 is a large-scale multi-target

agricultural pest detection dataset established for precision agriculture applications [17]. It contains 25,378 field pest images with a resolution of 800×600 pixels, which were collected using automatic pest trapping and imaging devices in agricultural environments. The dataset covers 24 typical field crop pest categories and contains 192,422 annotated pest instances in total.

The class-wise numbers of images and annotated instances in the Pest24 dataset are summarized in Table 3.

Table 3. Numbers of images and annotated instances for each pest category in Pest24.

Index	Pest Name	Images	Instances	Index	Pest Name	Images	Instances
1	Rice planthopper	316	1511	15	<i>Spodoptera cabbage</i>	1707	2302
2	Rice Leaf Roller	944	1240	16	<i>Scotogramma trifolii</i>	3223	4679
3	<i>Chilo suppressalis</i>	454	1285	24	Rottemberg	1388	1686
5	Armyworm	3828	8880	25	Yellow tiger	369	475
6	Bollworm	9049	28,014	28	Land tiger	154	168
7	Meadow borer	5526	16,516	29	Eight-character tiger	90	108
8	<i>Athetis lepigone</i>	7520	30,339	31	<i>Holotrichia oblita</i>	3111	11,675
10	<i>Spodoptera litura</i>	1588	1951	32	<i>Holotrichia parallela</i>	5228	53,347
11	<i>Spodoptera exigua</i>	3614	7263	34	<i>Anomala corpulenta</i>	3629	6528
12	Stem borer	1357	1804	35	<i>Gryllotalpa orientalis</i>	118	167
13	Little Gecko	2503	4279	36	Nematode trench	1814	6484
14	<i>Plutella xylostella</i>	531	953	37	<i>Agriotes fuscicollis</i>	239	768
					<i>Miwa</i>		
					Melahotus		

As shown in Table 3, Pest24 exhibits obvious class imbalance. For example, *Anomala corpulenta* contains 53,347 annotated instances, whereas *Holotrichia oblita* contains only 108 instances. In addition, many pest targets in Pest24 are small and appear in complex trap backgrounds, which increases the difficulty of accurate detection. Therefore, Pest24 provides a challenging external dataset for evaluating the generalization and robustness of FCD-DETR.

2.2. Baseline Model: RT-DETR-R18

RT-DETR is an end-to-end Detection Transformer designed to balance detection accuracy and inference efficiency [14]. It directly predicts object categories and bounding boxes without conventional post-processing procedures such as candidate proposal generation and non-maximum suppression, thereby balancing detection accuracy and inference efficiency. In this study, RT-DETR-R18 was selected as the baseline because its compact architecture is suitable for real-time multi-class pest detection in field monitoring scenarios.

RT-DETR-R18 consists of a backbone, a hybrid encoder, and a Transformer decoder. The backbone extracts multi-scale visual features, the hybrid encoder strengthens cross-scale feature interaction and semantic representation, and the decoder predicts object classes and bounding boxes from object queries.

Nevertheless, directly applying RT-DETR-R18 to UV insect-trap images remains limited because pest foregrounds are often mixed with dense targets, occlusion, stains, insect fragments, non-target insects, and morphologically similar species. These complex visual factors make it difficult for the baseline model to consistently focus on valid pest regions. These factors can weaken foreground–background discrimination, multi-scale feature fusion, and contextual modeling. Therefore, FCD-DETR introduces improvements in backbone feature extraction, encoder-level contextual modeling, and multi-scale feature fusion to enhance detection in complex UV insect-trap images.

2.3. Overall Architecture of FCD-DETR

FCD-DETR is built on the RT-DETR-R18 framework, and its overall architecture is shown in Figure 3. The network retains the end-to-end detection pipeline of RT-DETR and consists of a backbone, an encoder, a multi-scale feature fusion stage, and a decoder. Compared with RT-DETR-R18, FCD-DETR introduces C2f_CFBLOCK in the backbone, DHSA in the encoder, and FADM before the decoder to enhance feature extraction, contextual modeling, and foreground-aware multi-scale fusion in UV insect-trap pest images.

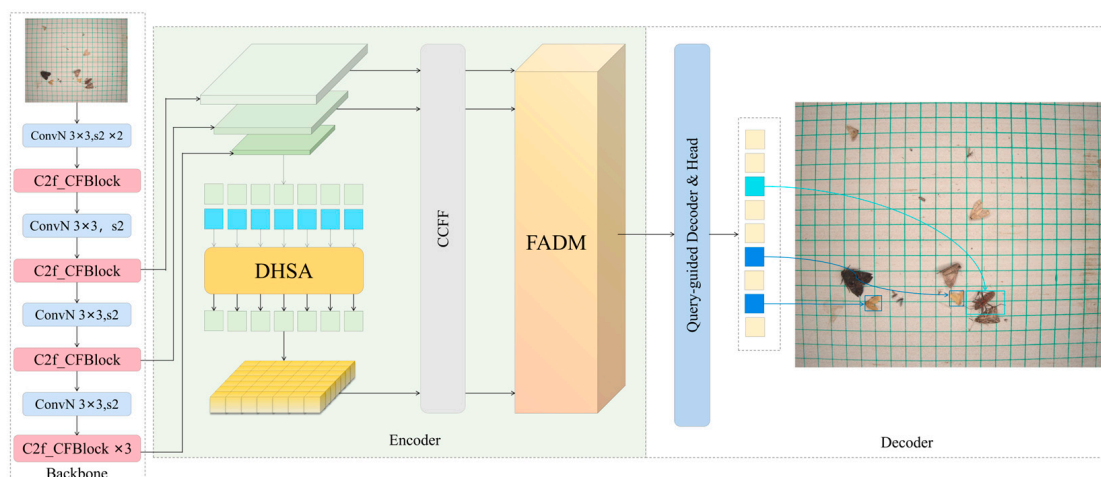


Figure 3. Overall architecture of FCD-DETR.

Given an input image, FCD-DETR first extracts hierarchical features using a C2f_CFBLOCK-based backbone. The extracted high-level features are then projected through channel mapping and fed into the DHSA-enhanced encoder to strengthen contextual interactions.

After encoder processing, top-down and bottom-up feature aggregation paths generate three fused feature maps, denoted as F3, F4, and F5. These features contain complementary spatial details and semantic information at different scales. FADM is then applied before the decoder to perform foreground-aware enhancement on these multi-scale features and suppress background interference.

Finally, the RT-DETR decoder uses object queries to decode the enhanced multi-scale features and predict pest classes and bounding boxes. By providing the decoder with foreground-enhanced and semantically enriched features, FCD-DETR improves detection robustness in dense and cluttered UV insect-trap scenes. Thus, FCD-DETR forms an improved detection framework that integrates backbone enhancement, encoder-level contextual modeling, and foreground-aware multi-scale fusion.

2.4. Lightweight Context-Enhanced Backbone with C2f_CFBLOCK

The backbone serves as the foundation for extracting hierarchical visual features in object detection models, and its feature representation capabilities directly influence the detection performance of the subsequent encoder and decoder. In images of pests captured by ultraviolet insect traps, pest targets often co-occur with background stains, insect remains, non-target insects, and pest categories with similar morphologies. Therefore, the backbone must not only preserve the local edges, textures, and morphological details of the pest targets but also possess a certain ability to model contextual features to enhance the discriminability between the target regions and the complex background.

C2f_CFBLOCK is related to existing context-enhanced backbone modules, which enhance feature representation by enlarging receptive fields or introducing contextual interactions. Similar to these methods, C2f_CFBLOCK improves contextual modeling; however,

its difference lies in embedding CFBBlock-style directional convolutional attention into a lightweight C2f structure. The learnable 7×1 and 1×7 convolutional attention branches capture horizontal and vertical contextual dependencies, while the convolutional MLP and residual connections help preserve local texture and contour cues. Therefore, C2f_CFBBlock combines lightweight C2f-style feature reuse with directional contextual modeling, making it more suitable for small, incomplete, and visually similar pests in cluttered UV insect-trap images.

The original C2f module achieves feature reuse and gradient flow through channel splitting, stacked bottleneck units, and feature fusion. However, its internal Bottleneck units mainly rely on conventional convolutions, which limits contextual modeling in complex UV insect-trap scenes. To address this, we replace the Bottleneck units in C2f with CFBBlock-based context-enhanced units [18], thereby constructing C2f_CFBBlock.

As shown in Figure 4, C2f_CFBBlock retains the original structure of C2f, which consists of 1×1 convolutional mapping, channel splitting, feature cascading, and final 1×1 convolutional fusion. Given the input feature X , channel mapping is first performed via 1×1 convolutions to obtain the feature X' . Subsequently, X' is split into two initial feature branches via a Split operation. One branch of features participates directly in subsequent fusion as a cross-layer connection, while the other branch sequentially passes through n CFBlocks for context-enhanced feature extraction. Finally, the features from each branch are concatenated along the channel dimension, and a 1×1 convolution is applied to generate the output feature, as shown in Equations (1) and (2):

$$X' = \text{Conv}_{1 \times 1}(X), [x_0, x_1] = \text{Split}(X'), x_{i+1} = \text{CFBlock}_i(x_i), i = 1, \dots, n \quad (1)$$

$$Y = \text{Conv}_{1 \times 1}(\text{Concat}(x_0, x_1, \dots, x_{n+1})) \quad (2)$$

Here, X represents the input features, X' represents the features after channel mapping, x_0 and x_1 represent the initial feature branches obtained after splitting, x_{i+1} represents the features obtained after passing through the i -th CFBBlock, and Y represents the output features of the C2f_CFBBlock.

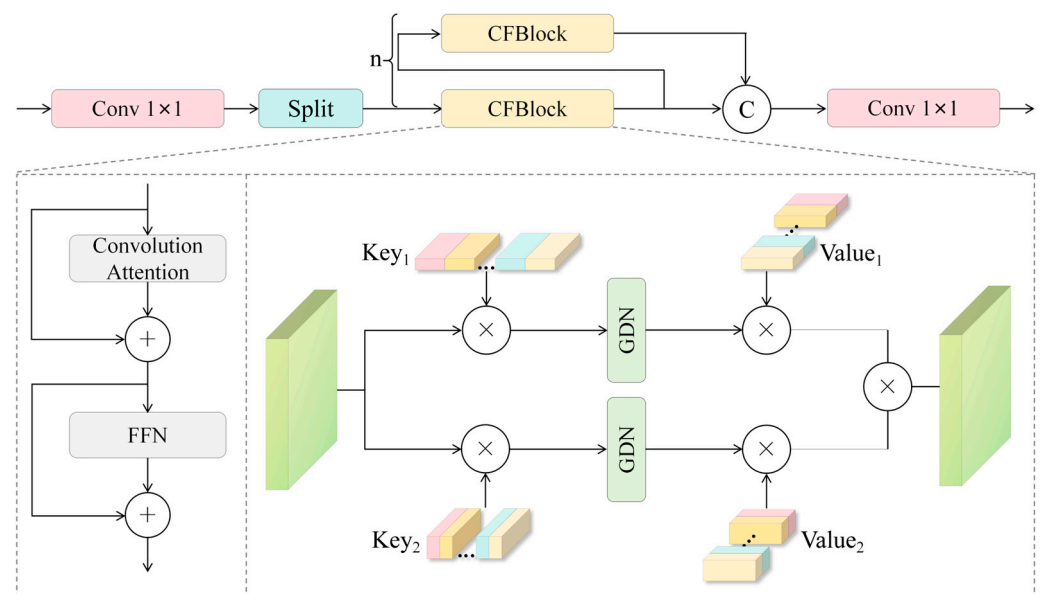


Figure 4. Structure of the C2f_CFBBlock module.

A CFBBlock consists of channel alignment, convolutional attention, an MLP, and a residual connection. When the number of input channels does not match the number of output channels, channel alignment is first performed via a 1×1 convolution. Subsequently,

the convolutional attention branch uses 7×1 and 1×7 convolutional kernels to model the features, enhancing spatial dependencies in both horizontal and vertical directions. The attention output is added to the input features via a residual connection and then undergoes nonlinear feature transformation through an MLP branch consisting of SyncBatchNorm, 3×3 convolutions, GELU, Dropout, and 3×3 convolutions. The residual computation process of CFBlock is expressed in Equation (3):

$$Z = X + \text{DropPath}(\text{CA}(X)), O = Z + \text{DropPath}(\text{MLP}(Z)) \quad (3)$$

Here, $\text{CA}(\cdot)$ denotes the convolutional attention operation, $\text{MLP}(\cdot)$ denotes the multilayer perceptron branch, Z denotes the intermediate features enhanced by attention, and O denotes the output features of the CFBlock.

In the backbone design of FCD-DETR, C2f_CFBlock is deployed across four feature extraction stages—P2, P3, P4, and P5—with corresponding channel counts of 128, 256, 384, and 384, respectively, where the P5 stage stacks three C2f_CFBlocks. This staged deployment enables the backbone to progressively capture hierarchical pest representations from low-level texture and contour details to high-level semantic information. By combining C2f-style feature branching and fusion with CFBlock-style directional contextual modeling, C2f_CFBlock strengthens the backbone representation while maintaining structural compactness. This design is beneficial for distinguishing valid pest regions from visually similar background interference in UV insect-trap images.

2.5. Dynamic-Range Histogram Self-Attention Encoder

The RT-DETR encoder is primarily used to enhance information interaction among high-level semantic features, providing more discriminative feature representations for subsequent multi-scale feature fusion and decoder prediction. In UV insect-trap images, different pest categories often exhibit similarities in wing shape, body posture, and texture, while background stains, insect fragments, and non-target insects frequently co-occur with target pests. These factors weaken the separability of high-level semantic features. If feature interaction relies only on raw spatial relationships, the encoder may struggle to distinguish pest-related semantic responses from visually similar interference.

DHSA is related to existing histogram attention mechanisms, which model feature relationships according to response distributions rather than only spatial positions. The similarity lies in that both DHSA and these methods reorganize feature responses to capture distribution-level dependencies. However, existing histogram attention mechanisms are mainly used in image restoration or general visual representation tasks, whereas DHSA is adapted to the RT-DETR encoder for pest detection. Its novelty lies in introducing dynamic-range response sorting and dual-branch histogram self-attention into the encoder, enabling high-level pest-related semantic features to interact at the response-distribution level. This design helps reduce semantic ambiguity among visually similar pests and background interference in cluttered UV insect-trap scenes.

In FCD-DETR, the high-level features from P5 are projected into channels and then fed into the DHSA-enhanced Transformer encoder layer. This architecture employs DHSA-based attention modeling to perform dynamic range histogram self-attention modeling on high-level semantic features [19], while retaining the residual connections, LayerNorm, and FFN structures from the Transformer encoder. By reorganizing feature responses according to their dynamic distribution, DHSA helps the encoder capture contextual dependencies beyond the original spatial arrangement. This encoder-level reconstruction enhances the model's ability to represent contextual relationships at the feature response distribution level while maintaining the stability of the original feature flow. As a result, the encoder can provide more robust semantic representations for subsequent multi-scale aggregation

and decoder prediction. This further improves the reliability of foreground-aware feature fusion and final pest localization in cluttered UV insect-trap scenes.

As shown in Figure 5, Attention_histogram first sorts a subset of the input feature channels along the spatial dimension to reorganize the feature response distribution. Subsequently, the sorted features undergo 1×1 convolutions and 3×3 deep convolutions to generate two sets of query/key representations and a value representation, namely q_1, k_1, q_2, k_2 , and v . Next, v is flattened and sorted by response intensity to obtain a dynamic response sorting index idx ; the same index is used to reorder q_1, k_1, q_2 , and k_2 , ensuring that attention calculations no longer rely solely on the original spatial order but instead establish contextual relationships at the level of feature response distributions. This process is represented by Equation (4):

$$q_1, k_1, q_2, k_2, v = DWConv(Conv_{1 \times 1}(F)), \hat{q}_1, \hat{k}_1, \hat{q}_2, \hat{k}_2, \hat{v} = Gather(q_1, k_1, q_2, k_2, v, idx) \quad (4)$$

Here, F denotes the encoder input features, $Conv_{1 \times 1}(\cdot)$ denotes channel mapping, $DWConv(\cdot)$ denotes depthwise convolution, idx denotes the index obtained by sorting the responses of v , $Gather(\cdot)$ denotes the rearrangement of features according to idx , and q_1, k_1, q_2, k_2 , and v denote the rearranged features.

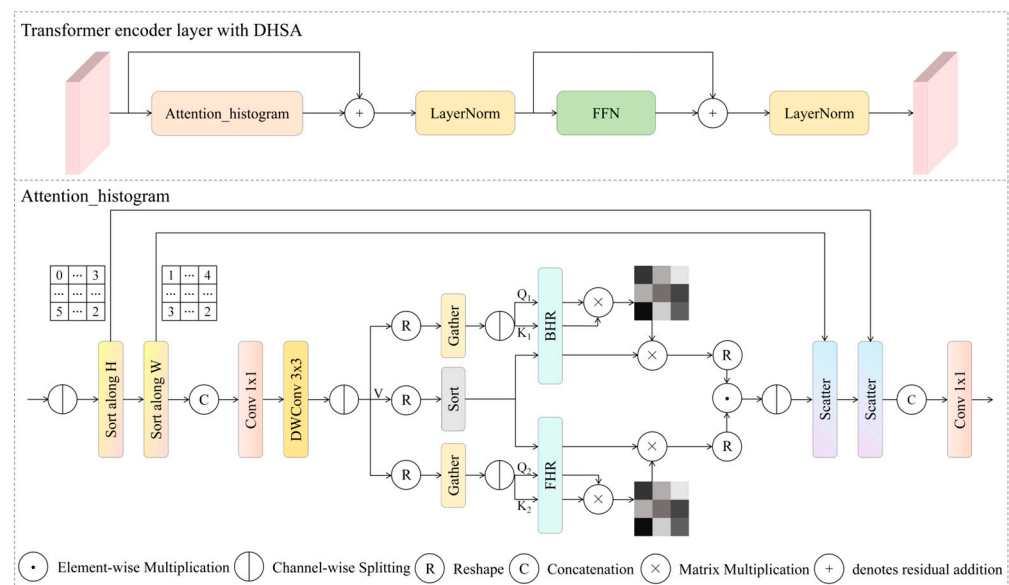


Figure 5. Structure of the DHSA-enhanced Transformer encoder layer.

The rearranged features are fed into two separate histogram attention branches. The two branches use different feature arrangements for attention computation, denoted as H_{BHR} and H_{FHR} , respectively, to model contextual relationships under different dynamic range organization schemes. Subsequently, the outputs of the two branches are restored to their original index order via $Scatter(\cdot)$, fused through element-wise multiplication, and finally projected via a 1×1 projection to obtain the enhanced encoder features. The enhanced encoder feature is obtained by Equation (5):

$$F_{dhsa} = Proj\left(Scatter\left(H_{BHR}\left(\hat{q}_1, \hat{k}_1, \hat{v}\right), idx\right) \odot Scatter\left(H_{FHR}\left(\hat{q}_2, \hat{k}_2, \hat{v}\right), idx\right)\right) \quad (5)$$

Here, $H_{BHR}(\cdot)$ and $H_{FHR}(\cdot)$ represent the histogram attention branches under two different rearrangement schemes, respectively; $Scatter(\cdot)$ restores the attention output to its original index order; $\odot$ denotes element-wise multiplication; $Proj(\cdot)$ denotes the output projection operation; and F_{dhsa} denotes the encoder output features enhanced by DHSA.

In the DHSA-enhanced Transformer encoder layer, F_{dhsa} is fused with the input features via a residual connection and normalized by LayerNorm; subsequently, the features are further fed into an FFN branch composed of 1×1 convolutions, and the final encoder-enhanced features are output via another residual connection and LayerNorm. Compared with attention mechanisms that rely mainly on spatial relationships, DHSA strengthens response-distribution-level semantic interaction in the encoder, providing more discriminative high-level features for subsequent multi-scale aggregation and FADM-based foreground-aware fusion.

2.6. Foreground-Aware Decoupled Mixer for Multi-Scale Feature Fusion

Multi-scale feature fusion is essential for addressing scale variations in object detection. Upon traversing the backbone, encoder, and the top-down and bottom-up feature aggregation layers, the network acquires feature maps that embody various spatial resolutions and semantic levels. In images obtained from UV insect traps, target pests frequently coexist alongside background stains, insect remnants, non-target insects, and adjacent insects. However, direct multi-scale fusion may mix pest foreground cues with background responses from trap stains, insect fragments, non-target insects, and adjacent insects, thereby weakening the foreground discriminability of the fused features. To address this problem, we propose the Foreground-Aware Decoupled Mixer (FADM), which performs foreground-aware multi-scale feature refinement before the RT-DETR decoder.

FADM utilizes the three-scale features F_3 , F_4 , and F_5 —derived from feature aggregation—as input to produce refined feature representations for each target scale individually. Figure 6 illustrates the comprehensive architecture of FADM, which predominantly comprises multi-scale feature alignment, a semantic branch, a foreground branch, decoupled fusion, and spatial-channel gating modulation.

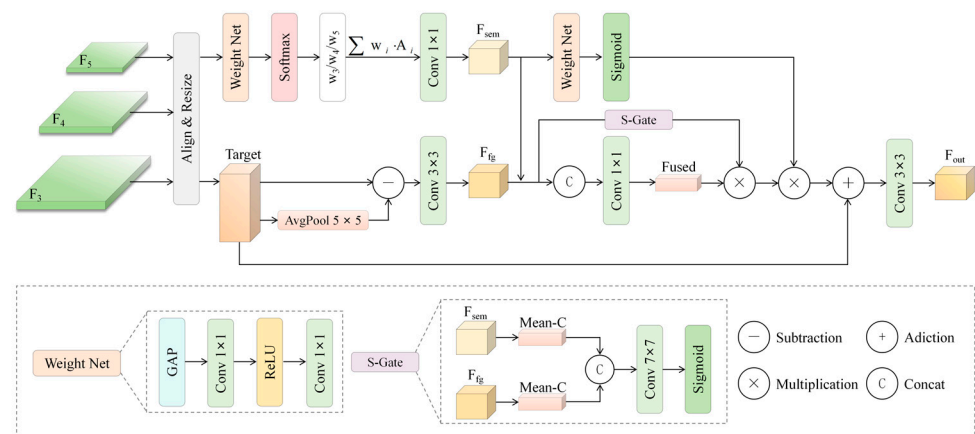


Figure 6. Architecture of the FADM module.

The main originality of FADM lies in its foreground-semantic decoupling design for multi-scale feature fusion. Existing foreground-aware or attention-guided fusion methods usually enhance features by directly concatenating, reweighting, or generating attention maps from fused multi-scale representations. In contrast, FADM explicitly separates foreground cue extraction from multi-scale semantic aggregation before the RT-DETR decoder. The semantic aggregation branch adaptively aggregates aligned multi-scale features to preserve cross-scale contextual semantics, whereas the foreground cue extraction branch extracts local foreground variations from the target-scale feature to emphasize pest-related structural cues. These two complementary branches are then recombined through decoupled fusion and spatial-channel gating. Therefore, FADM is not a simple attention-based reweighting module, but a detection-oriented foreground-semantic decoupling

mechanism designed for dense, occluded, scale-varied, and background-contaminated UV insect-trap pest detection.

For target scales $l \in 3, 4, 5$, let T_l represent the target feature at each scale subsequent to channel mapping. FADM initially maps information from various scales to a standardized channel dimension using 1×1 convolutions and aligns them according to the spatial dimensions of T_l , as defined in Equation (6):

$$\tilde{F}_{i,l} = \text{Resize}(\text{Conv}_{1 \times 1}(F_i), T_l), i \in 3, 4, 5 \quad (6)$$

In the equation, F_i symbolizes the input feature at the i -th scale, while $F_{i,l}$ signifies the feature post-alignment to the target scale T_l ; $\text{Resize}(\cdot)$ indicates spatial resizing. Post-alignment, the features across the three scales possess identical channel counts and spatial dimensions, establishing a basis for later cross-scale information integration.

In the semantic domain, FADM employs a Weight Net to assess the contribution of features across various scales to the present target scale. The Weight Net comprises global average pooling, 1×1 convolutions, ReLU activation, and additional 1×1 convolutions, utilized for the generation of scale-adaptive weights. Thereafter, the weights for the three scales are normalized using Softmax along the scale dimension and employed to weight-blend the aligned multi-scale features, resulting in the semantic feature $F_{sem,l}$, as computed in Equation (7):

$$w_{i,l} = \text{Softmax}_i\left(W\left(\tilde{F}_{i,l}\right)\right), F_{sem,l} = \text{Conv}_{1 \times 1}\left(\sum_{i=3}^5 w_{i,l} \cdot \tilde{F}_{i,l}\right) \quad (7)$$

In this context, $W(\cdot)$ signifies the Weight Net, and $w_{i,l}$ indicates the adaptive weight of the i -th scale feature at the target scale l .

In contrast to the semantic branch, the foreground branch predominantly emphasizes local salient reactions within target-scale features. To achieve the goal feature T_l , FADM utilizes 5×5 average pooling to generate a locally smoothed response and captures local response variations via a difference operation. The outcome of the difference is next processed by a 3×3 convolution to derive the foreground feature $F_{fg,l}$, as defined in Equation (8):

$$F_{fg,l} = \text{Conv}_{3 \times 3}(T_l - \text{AvgPool}_{5 \times 5}(T_l)) \quad (8)$$

This difference operation highlights local variations such as pest edges, wing patterns, and body contours, while suppressing relatively smooth background responses caused by stains, trap textures, and fragments.

Thereafter, FADM merges the foreground feature $F_{fg,l}$ with the semantic feature $F_{sem,l}$ along the channel dimension and executes decoupled fusion using a 1×1 convolution to get fused feature $F_{fuse,l}$. To enhance identification of effective pest locations and mitigate background interference, FADM implements spatial gating S_l and channel gating C_l . The spatial gate is produced by implementing a 7×7 convolution followed by Sigmoid on the channel averages of $F_{fg,l}$ and $F_{sem,l}$, serving to emphasize critical spatial locations; the channel gate is created through global average pooling and two layers of 1×1 convolutions applied to $F_{sem,l}$, aimed at augmenting the discriminative semantic channel. The gated-modulated fused features are ultimately included into the target-scale features using residual summation, yielding the output features via a 3×3 convolution, as formulated in Equation (9).

$$F_{fuse,l} = \text{Conv}_{1 \times 1}\left(\text{Concat}\left(F_{fg,l}, F_{sem,l}\right)\right), F_{out,l} = \text{Conv}_{3 \times 3}\left(T_l + F_{fuse,l} \odot S_l \odot C_l\right) \quad (9)$$

In the equation, $\odot$ signifies element-wise multiplication, whereas S_l and C_l denote the spatial gating weights and channel gating weights, respectively. FADM is applied to the three target scales F_3 , F_4 , and F_5 , respectively, to generate enhanced multiscale features, which are then fed into the RT-DETR decoder for class and bounding box prediction.

3. Results

3.1. Experimental Environment and Parameter Settings

All experiments were conducted on a Linux server equipped with an NVIDIA GeForce RTX 4090 GPU. The software environment included Python 3.10.14, PyTorch 2.2.2 and CUDA 11.8. For fair comparison, all models were trained and evaluated using the same dataset split, input resolution, and training configuration. The input size was set to 640×640 pixels, the batch size to 32, and the training epoch to 100. AdamW [20] was used as the optimizer with an initial learning rate of 0.0001.

3.2. Evaluation Metrics

To evaluate detection performance and computational efficiency on the UV insect-trap pest dataset, we used *Precision*, *Recall*, *mAP@0.5*, *mAP@50:95*, *F1-score*, *FPS*, and *Params*. *Precision* and *Recall* measure the correctness and completeness of pest detection, respectively. *mAP@0.5* evaluates category-level average precision at an IoU threshold of 0.5, whereas *mAP@50:95* averages AP over IoU thresholds from 0.5 to 0.95 with a step size of 0.05, providing a stricter assessment of localization accuracy following the COCO evaluation protocol [21]. *F1-score* measures the balance between *Precision* and *Recall*, and *Params* denotes model complexity.

The formulas for *Precision* and *Recall* are shown in Equations (10) and (11):

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

where *TP* represents the number of pest targets correctly detected, *FP* represents the number of targets incorrectly detected as pests, and *FN* represents the number of true pest targets not detected by the model.

The *F1-score* is the harmonic mean of *Precision* and *Recall*, as calculated in Equation (12):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

Average Precision (*AP*) represents the area under the *Precision–Recall* curve, as defined in Equation (13):

$$AP = \int_0^1 P(R) dR \quad (13)$$

mAP represents the average of the *AP*s across all categories, as computed in Equation (14):

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (14)$$

Here, *N* denotes the number of pest categories, with $N = 36$ in this study, and AP_i denotes the average precision of the *i*-th category. Higher *Precision*, *Recall*, *mAP*, and *F1-score* values indicate better detection performance, whereas fewer *Params* indicate a more compact model.

3.3. Comparative Analysis of Different Detection Models

3.3.1. Comparison of Evaluation Indicators for Different Models

To evaluate the effectiveness of FCD-DETR, we compared it with Faster R-CNN [22], representative YOLO models [23,24], D-FINE-M [25], and RT-DETR variants [14] under the same experimental settings. The quantitative results are reported in Table 4.

Table 4. Quantitative comparison of detection models on the UV insect-trap pest dataset.

Model	Precision	Recall	mAP@0.5	mAP@50:95	F1	FPS	Params
Faster R-CNN	35.06	59.84	26.77	12.50	44.22	93.57	41.48
YOLOv5m	60.12	55.21	59.13	37.34	54.88	348.24	25.07
YOLOv8m	60.21	55.84	60.56	37.96	56.00	311.61	25.86
YOLOv9m	64.57	53.32	59.95	37.21	54.95	343.22	20.04
YOLOv10m	57.84	56.95	57.21	37.62	54.33	269.24	15.33
YOLOv11m	57.91	57.46	59.48	37.94	54.63	369.57	20.06
D-FINE-M	74.03	67.49	53.40	32.50	70.61	286.75	19.00
RT-DETR-L	58.25	62.02	60.03	37.81	57.92	189.16	32.06
RT-DETR-R34	60.95	59.36	59.93	37.42	58.21	240.82	31.16
RT-DETR-R18	60.14	59.98	59.17	36.22	58.17	272.55	19.90
FCD-DETR	61.66	63.84	64.42	41.33	61.32	224.95	20.79

As shown in Table 4, Faster R-CNN performs substantially worse than the other models, achieving only 26.77% *mAP@0.5* and 12.50% *mAP@50:95*. Among the YOLO models, YOLOv8m obtains the best overall performance, with 60.56% *mAP@0.5* and 37.96% *mAP@50:95*, whereas YOLOv9m achieves the highest *Precision* but lower *Recall*, indicating a higher risk of missed detections.

For the RT-DETR variants, RT-DETR-L and RT-DETR-R34 outperform RT-DETR-R18 but require 32.06M and 31.16 M parameters, respectively. In contrast, FCD-DETR achieves the highest *Recall*, *mAP@0.5*, *mAP@50:95*, and *F1*-score, reaching 63.84%, 64.42%, 41.33%, and 61.32%, respectively, with only 20.79 M parameters. Compared with RT-DETR-R18, FCD-DETR improves *mAP@0.5* and *mAP@50:95* by 5.25 and 5.11 percentage points; compared with YOLOv8m, it improves these metrics by 3.86 and 3.37 percentage points. In terms of inference speed, FCD-DETR achieves 224.95 FPS, which is lower than the RT-DETR-R18 baseline due to the introduced feature enhancement modules, but still maintains a relatively high inference speed while substantially improving detection accuracy.

These results indicate that FCD-DETR achieves a better accuracy–complexity–efficiency trade-off than the compared models and improves localization quality under stricter IoU thresholds.

3.3.2. Overall Precision-Recall Curve Comparison

To further evaluate the detection performance of different models across varying confidence thresholds, we plotted the overall *Precision–Recall* curves of representative models, as shown in Figure 7. To avoid reduced readability caused by too many curves, we selected YOLOv8m and YOLOv9m as the two strongest CNN-based comparison models, and RT-DETR-L and RT-DETR-R34 as the two strongest Transformer-based comparison models. The RT-DETR-R18 baseline and the proposed FCD-DETR were also included for comparative analysis.

As shown in Figure 7, the *Precision–Recall* curve of FCD-DETR is generally located above those of the other models, achieving an *mAP@0.5* of 0.644. This value is higher than those of YOLOv8m, YOLOv9m, RT-DETR-L, RT-DETR-R34, and RT-DETR-R18, which achieve 0.606, 0.600, 0.600, 0.599, and 0.592, respectively. Compared with the RT-DETR-R18

baseline, FCD-DETR maintains higher *Precision* over most *Recall* ranges, indicating that the improved model can better control false positives while enhancing pest target detection.

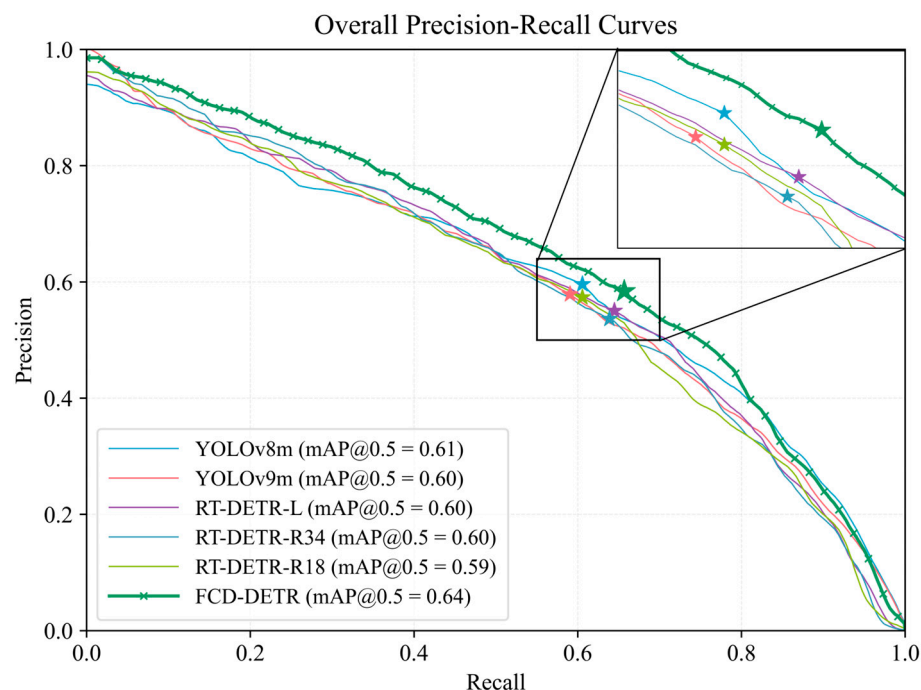


Figure 7. Comprehensive Precision–Recall curves of exemplary detection methods. Colored stars indicate the maximum F1-score points.

From the perspective of curve variation, all models maintain relatively high *Precision* when *Recall* is in the low-to-middle range. However, the FCD-DETR curve decreases more gradually overall, indicating more stable detection performance under different confidence thresholds. As *Recall* gradually increases, the *Precision* of the other models decreases more noticeably, whereas FCD-DETR still maintains relatively high *Precision* across a wider *Recall* range.

3.3.3. Visual and Quantitative Analysis of Detection Results

To further visually compare the detection performance of different models on real UV insect-trap pest images, representative samples were selected for analysis. The comparison includes the ground truth, RT-DETR-R18 baseline, YOLOv8m, RT-DETR-L, and FCD-DETR, with red circles indicating false positive targets and purple circles indicating missed targets. These samples cover typical challenging conditions in the dataset, including sparse targets, dense distributions, background grids, insect fragments, and low-contrast pest regions.

As shown in Figure 8, in scenes with fewer targets and relatively simple backgrounds, all models can detect prominent pest targets, but false positives and missed detections still occur to varying degrees. In scenes with clear background grids, numerous insect fragments, and dense pest distributions, the differences among the models become more evident. The RT-DETR-R18 baseline shows insufficient detection of some small-scale and adjacent pest targets. YOLOv8m obtains relatively accurate bounding boxes in some images, but still misses low-contrast and densely distributed targets. Although RT-DETR-L shows better overall detection capability than the baseline, it still suffers from missed detections in complex backgrounds and edge target regions. These results indicate that increasing model capacity alone does not fully resolve the challenges caused by cluttered backgrounds and dense pest distributions.

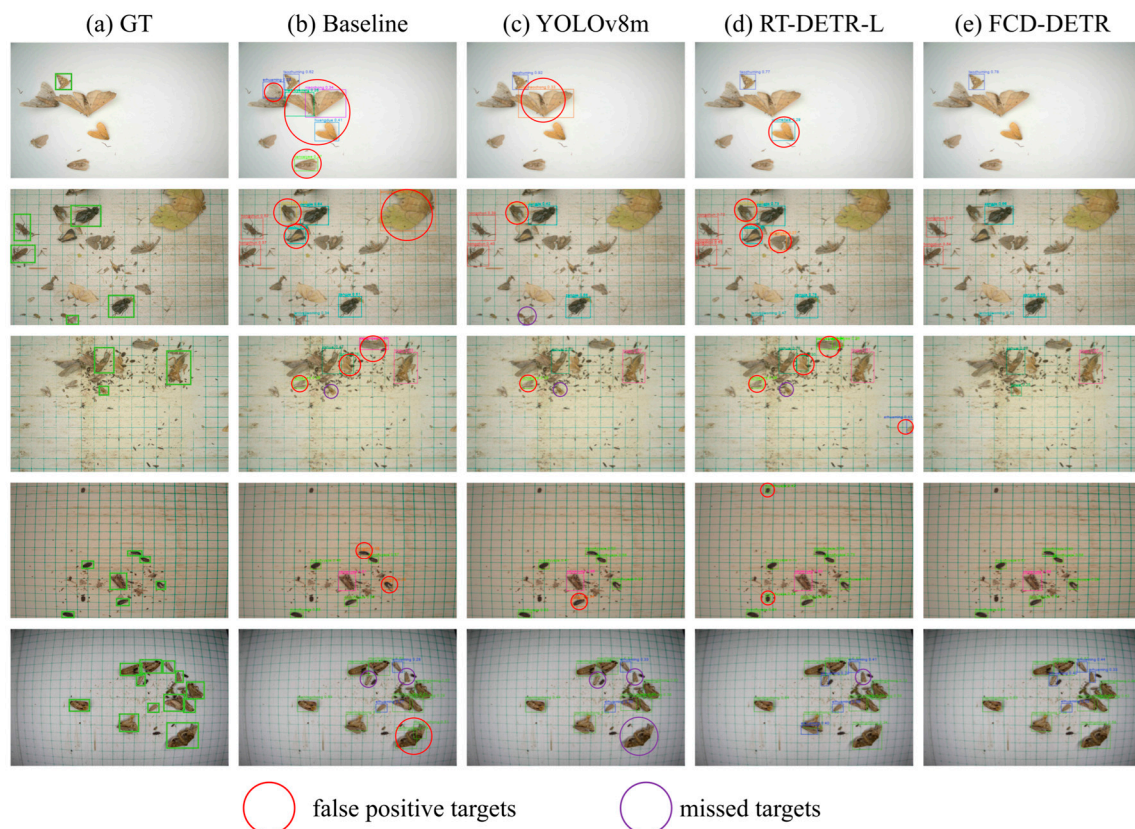


Figure 8. Visual comparison of detection results in representative light-trap pest images. (a) Ground truth; (b) RT-DETR-R18 baseline; (c) YOLOv8m; (d) RT-DETR-L; (e) FCD-DETR. Red circles indicate false positive targets, and purple circles indicate missed targets.

In contrast, FCD-DETR shows better detection completeness in most representative samples. For pest targets with dense distributions, obvious scale variations, and strong background interference, FCD-DETR detects more valid targets while reducing false positives and missed detections. This improvement indicates that the proposed foreground-aware fusion strategy helps the model focus more effectively on true pest regions. Its predicted bounding boxes are more consistent with the ground-truth annotations, indicating more stable target localization in complex light-trap environments.

Overall, the visual comparison in Figure 8 shows that FCD-DETR achieves detection results that are closer to the ground-truth annotations than the compared models. It performs more stably in challenging scenarios involving small-scale targets, adjacent targets, and background interference, indicating stronger robustness against background clutter, target adhesion, and scale variation. However, visual comparison alone provides mainly qualitative evidence and cannot directly quantify the reduction in false positives and missed detections. Therefore, to provide a more objective assessment of the visual results, we further analyzed the false positives (FPs) and false negatives (FNs) of representative pest categories in the test set. The results are shown in Table 5.

As shown in Table 5, FCD-DETR reduces both false-positive and false-negative detections for the selected representative categories. Overall, the number of false positives decreases from 300 to 211, corresponding to a reduction of 29.7%, while the number of false negatives decreases from 155 to 120, corresponding to a reduction of 22.6%. These results provide more objective evidence that FCD-DETR improves foreground discrimination and reduces missed detections in complex UV insect-trap images.

Table 5. FP and FN reductions for representative pest categories.

Pest Category	GT	RT-DETR FP	FCD-DETR FP	FP Reduction	RT-DETR FN	FCD-DETR FN	FN Reduction
C6	30	30	25	16.7%	6	3	50.0%
C9	37	48	28	41.7%	23	16	30.4%
C18	144	51	47	7.8%	52	37	28.8%
C27	129	54	28	48.1%	46	42	8.7%
C32	287	117	83	29.1%	28	22	21.4%

3.4. Ablation Experiments

By comparing single-module variants, pairwise combinations, and the complete model, the ablation study provides a clear evaluation of the individual effectiveness of each module and their synergistic contributions to foreground-aware feature enhancement, contextual semantic modeling, and multi-scale feature fusion.

As shown in Table 6, each module individually improves $mAP@0.5$ and $mAP@50:95$ over the RT-DETR-R18 baseline. FADM increases $mAP@0.5$ and $mAP@50:95$ from 59.17% and 36.22% to 60.46% and 37.32%, respectively, indicating improved foreground-aware fusion. DHSA achieves the highest *Precision* among the single-module variants, reaching 66.70%, with $mAP@0.5$ and $mAP@50:95$ increasing to 61.75% and 38.09%. C2f_CFBLOCK improves $mAP@0.5$ and $mAP@50:95$ to 61.79% and 37.54%, while reducing the parameter count from 19.90M to 16.32M.

Table 6. Ablation results of FCD-DETR.

Baseline	FADM	DHSA	C2f_CFBLOCK	Precision	Recall	$mAP@0.5$	$mAP@50:95$	F1	FPS	Params
✓	×	×	×	60.14	59.98	59.17	36.22	58.17	272.55	19.90
✓	✓	×	×	63.26	60.10	60.46	37.32	59.50	246.85	24.25
✓	×	✓	×	66.70	59.50	61.75	38.09	60.47	377.13	20.06
✓	×	×	✓	63.76	59.44	61.79	37.54	59.10	319.91	16.32
✓	✓	✓	×	64.29	60.35	62.44	38.91	60.35	233.78	24.39
✓	×	✓	✓	61.91	62.97	63.65	40.10	60.33	209.17	16.46
✓	✓	×	✓	62.95	60.81	63.40	40.12	60.18	248.33	20.65
✓	✓	✓	✓	61.66	63.84	64.42	41.33	61.32	224.95	20.79

Pairwise combinations further improve performance, suggesting complementary effects among the modules. The combination of FADM and DHSA achieves 62.44% $mAP@0.5$ and 38.91% $mAP@50:95$. The combination of DHSA and C2f_CFBLOCK achieves 63.65% $mAP@0.5$ and 40.10% $mAP@50:95$ with only 16.46M parameters, while FADM combined with C2f_CFBLOCK reaches 63.40% $mAP@0.5$ and 40.12% $mAP@50:95$.

With all three modules integrated, FCD-DETR achieves the best overall performance, with 64.42% $mAP@0.5$ and 41.33% $mAP@50:95$, improving the baseline by 5.25 and 5.11 percentage points, respectively. *Recall* and *F1*-score also increase from 59.98% and 58.17% to 63.84% and 61.32%, respectively, while the parameter count increases only slightly from 19.90 M to 20.79 M. Meanwhile, FCD-DETR maintains 224.95 FPS, showing that the complete model improves accuracy with an acceptable efficiency cost. These results confirm that the three modules jointly improve detection accuracy and localization quality with limited additional complexity.

3.5. Comparison with Alternative Improvement Modules

3.5.1. Comparison of Backbone Enhancement Modules

We assessed the efficacy of C2f_CFBLOCK for backbone enhancement by comparing it with C2f, C2f_Ghost [26], C2f_CBAM [27], and C2f_CA [28] under identical replacement conditions in RT-DETR-R18. The findings are presented in Table 7.

Table 7. Evaluation of backbone augmentation modules.

Model	Precision	Recall	mAP@0.5	mAP@50:95	F1	FPS	Params
RT-DETR-R18	60.14	59.98	59.17	36.22	58.17	272.55	19.90
+C2f	60.30	60.11	59.00	36.21	58.16	295.12	15.46
+C2f_Ghost	60.44	57.73	58.51	35.89	56.95	298.10	12.53
+C2f_CBAM	59.65	61.22	60.31	37.07	58.28	282.81	15.48
+C2f_CA	60.86	59.37	59.85	36.59	57.62	277.37	15.48
+C2f_CFBLOCK	63.76	59.44	61.79	37.54	59.10	319.91	16.32

Table 7 illustrates that C2f and C2f_Ghost decrease the parameter count while yielding minimal enhancements in average precision (AP). C2f_Ghost induces a decline in accuracy, indicating that an overly lightweight design may compromise feature representation. Despite the enhancement in detection performance due to the incorporation of attention mechanisms in C2f_CBAM and C2f_CA, their AP-based metrics continue to be inferior to those of C2f_CFBLOCK.

C2f_CFBLOCK demonstrates superior overall performance among the evaluated backbone modules, attaining 61.79% *mAP@0.5* and 37.54% *mAP@50:95*. The parameter count is 16.32 million, which is less than the 19.90 million of the RT-DETR-R18 baseline, suggesting that C2f_CFBLOCK enhances accuracy while preserving a compact architecture.

The results indicate that C2f_CFBLOCK offers a superior accuracy-complexity-efficiency trade-off compared to basic lightweight modules and general attention-based alternatives. The integration of C2f-style feature branching with CFBLOCK-based contextual modeling renders it more appropriate for backbone augmentation in FCD-DETR.

3.5.2. Comparison of Encoder Attention Modules

To assess the efficacy of DHSA in encoder-level semantic interaction, we compared it with ASSA [29], DAttention [30], HiLo [31], and PolaFormer [32] under identical replacement conditions in the RT-DETR-R18 encoder. The findings are presented in Table 8.

Table 8. Comparative Analysis of Encoder Attention Modules.

Model	Precision	Recall	mAP@0.5	mAP@50:95	F1	FPS	Params
RT-DETR-R18	60.14	59.98	59.17	36.22	58.17	272.55	19.90
+ASSA	60.98	60.15	60.24	37.98	58.19	290.27	20.76
+DAttention	63.02	58.26	59.32	37.08	57.97	291.02	19.92
+HiLo	58.92	57.81	58.28	36.51	56.61	306.29	19.88
+PolaFormer	61.43	58.63	58.97	36.28	57.99	296.65	20.09
+DHSA	66.70	59.50	61.75	38.09	60.47	377.13	20.06

Table 8 illustrates that the evaluated attention modules enhance detection performance to varying degrees; nonetheless, none exceed the efficacy of DHSA. ASSA enhances AP measures to a certain extent; DAttention achieves 63.02% *Precision* but offers minimal AP enhancement, while HiLo and PolaFormer continue to underperform compared to DHSA in overall detection efficacy. The results indicate that traditional attention strategies

may inadequately mitigate background interference and inter-class similarity in UV insect-trap images.

DHSA attains superior overall performance, exhibiting 66.70% *Precision*, 61.75% *mAP@0.5*, 38.09% *mAP@50:95*, and 60.47% *F1*-score. The parameter count is 20.06M, which is lower than that of ASSA, suggesting that DHSA enhances detection performance without significantly increasing model complexity. In addition, DHSA achieves the highest FPS among the compared attention modules, indicating that its encoder-level enhancement improves detection accuracy while maintaining efficient inference.

In general, DHSA offers a superior accuracy-complexity balance compared to alternative attention modules. By modeling contextual relationships at the feature-response distribution level through dynamic response ordering and dual-branch histogram attention, DHSA effectively differentiates pest targets from intricate backgrounds and morphologically analogous categories, rendering it appropriate for encoder enhancement in FCD-DETR.

3.5.3. Comparison of Multi-Scale Feature Fusion Modules

To assess the efficacy of FADM in multi-scale feature fusion, we conducted a comparison with notable fusion modules, namely BiFPN [33] and BFP [34], under identical replacement conditions in RT-DETR-R18. The findings are presented in Table 9.

Table 9. Comparative Analysis of Multi-Scale Feature Fusion Modules.

Model	Precision	Recall	mAP@0.5	mAP@50:95	F1	FPS	Params
RT-DETR-R18	60.14	59.98	59.17	36.22	58.17	272.55	19.90
+BiFPN	60.12	58.77	59.69	37.27	57.18	279.99	21.69
+BFP	61.66	56.98	58.31	35.78	57.05	275.62	21.69
+FADM	63.26	60.10	60.46	37.32	59.50	246.85	24.25

Table 9 demonstrates that BiFPN enhances *mAP@0.5* and *mAP@50:95*, although it decreases *Precision*, *Recall*, and *F1*-score in comparison to the baseline. BFP enhances *Precision* to 61.66%, however its *Recall* and *AP*-based measures deteriorate. The results demonstrate that traditional weighted or balanced fusion procedures do not reliably enhance detection performance in intricate UV insect-trap images.

FADM attains superior overall performance among the fusion modules, exhibiting 63.26% *Precision*, 60.10% *Recall*, 60.46% *mAP@0.5*, 37.32% *mAP@50:95*, and 59.50% *F1*-score. In comparison to RT-DETR-R18, FADM enhances *Precision*, *mAP@0.5*, and *mAP@50:95* by 3.12, 1.29, and 1.10 percentage points, respectively. In terms of inference speed, FADM achieves 246.85 FPS, indicating that its accuracy improvement is accompanied by a moderate reduction in inference speed.

FADM offers a more effective fusion approach by enhancing foreground regions and diminishing background interference. Despite introducing supplementary parameters, its accuracy improvements are more significant, reinforcing its function as the primary multi-scale feature fusion module in FCD-DETR.

3.6. Per-Class Detection Performance Analysis

We assessed category-level detection ability by comparing the per-class *AP@0.5* of RT-DETR-R18 and FCD-DETR, as illustrated in Figure 9. This comparison helps reveal whether the overall performance improvement is broadly distributed across pest categories or mainly contributed by a few dominant classes.

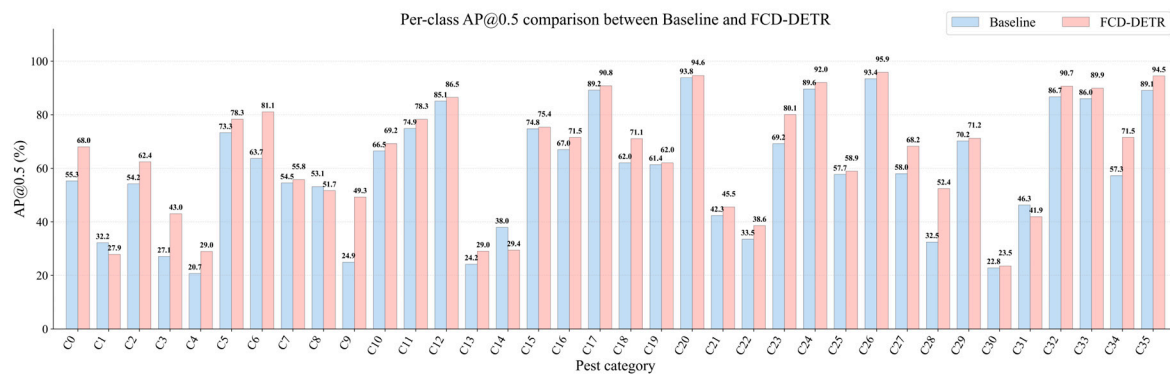


Figure 9. Comparison of per-class AP@0.5 between RT-DETR-R18 and FCD-DETR.

Figure 9 illustrates that FCD-DETR enhances AP@0.5 in 32 out of 36 pest categories, yielding an average category-level increase of around 5.24 percentage points. This signifies that the suggested model enhances both overall detection accuracy and category-level detection stability.

The improvements are particularly evident in several low-frequency or low-AP categories. For example, C28 contains only 86 instances, but its AP@0.5 increases from 32.5% to 52.4%; C9 also improves from 24.9% to 49.3%. These results suggest that FCD-DETR can help alleviate the adverse influence of long-tailed class distribution to some extent by enhancing foreground discrimination and semantic representation.

For categories with elevated baseline AP@0.5, including C20, C24, C26, C32, C33, and C35, FCD-DETR sustains or enhances performance, demonstrating consistent recognition of well-established categories. Nonetheless, AP@0.5 diminishes for several categories, specifically C1, C8, C14, and C31. The reductions may be attributed to inter-class similarities in wing morphology, body pigmentation, and texture, ambiguous target delineations, insufficient discriminative characteristics, and a scarcity of data in low-frequency categories.

FCD-DETR enhances category-level recognition for the majority of pest classes; nonetheless, morphologically similar, weak-textured, and low-frequency categories continue to pose challenges in UV insect-trap pest detection. These results suggest that future improvements should further focus on fine-grained feature learning and long-tailed category optimization.

3.7. Grad-CAM Visualization Analysis

We employed Grad-CAM [35] to illustrate the discriminative regions utilized by various models, including RT-DETR-R18, YOLOv8m, RT-DETR-L, and FCD-DETR, as depicted in Figure 10. Enhanced Grad-CAM responses signify areas that significantly influence model predictions.

As shown in Figure 10, RT-DETR-R18, YOLOv8m, and RT-DETR-L show relatively dispersed Grad-CAM responses in some complex samples, with activations extending to background grids, stains, insect fragments, or non-target regions. In contrast, FCD-DETR produces more concentrated responses on pest-related foreground areas and reduces activations in irrelevant background regions. These results indicate that the proposed modules help the model focus more effectively on discriminative pest structures and improve robustness to background interference.

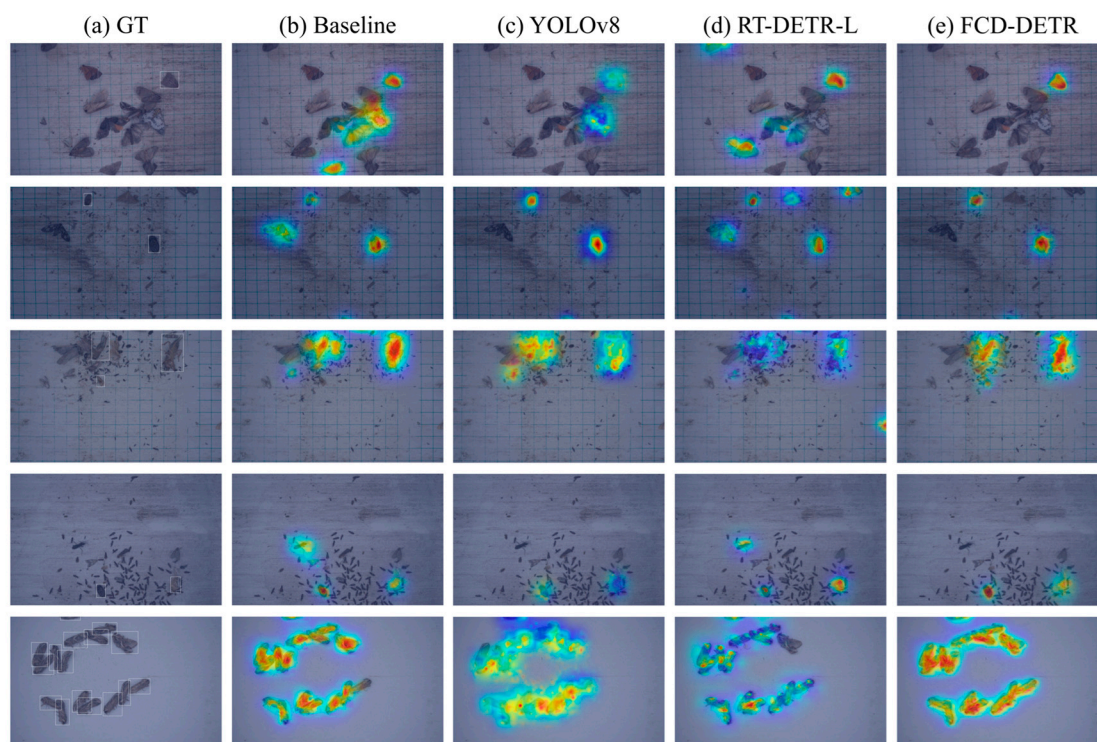


Figure 10. Comparative visualization of Grad-CAM across representative detection models. (a) Ground truth; (b) Baseline; (c) YOLOv8m; (d) RT-DETR-L; (e) FCD-DETR.

3.8. Confusion Matrix Analysis

To assess class-level discrimination in multi-class pest detection, we generated normalized confusion matrices for RT-DETR-R18 and FCD-DETR, illustrated in Figures 11a and 11b, respectively. In each matrix, the horizontal axis denotes the true class, the vertical axis signifies the predicted class, diagonal values reflect accurately identified proportions, and off-diagonal values suggest inter-class confusion.

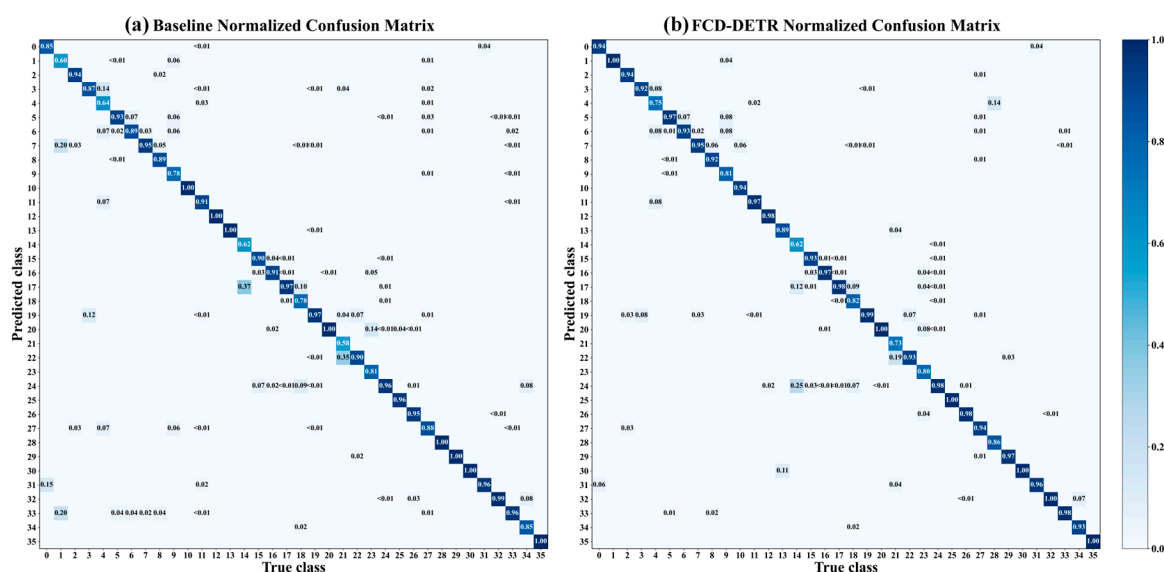


Figure 11. Normalized confusion matrix comparison between the RT-DETR-R18 baseline and FCD-DETR. (a) RT-DETR-R18 baseline; (b) FCD-DETR.

Figure 11a illustrates that RT-DETR-R18 has significant off-diagonal responses for many categories, signifying misclassification across adjacent or morphologically analogous

pest classes. In contrast, Figure 11b illustrates a more pronounced diagonal response for FCD-DETR, indicating elevated correct recognition rates across the majority of categories. The accurate recognition rate escalates from 0.60 to 1.00 for C1, from 0.64 to 0.75 for C4, from 0.72 to 0.81 for C9, from 0.62 to 0.77 for C21, and from 0.85 to 0.93 for C34.

The off-diagonal regions further show that FCD-DETR reduces several confusion patterns observed in the baseline model. RT-DETR-R18 exhibits more ambiguity between categories C14 and C17, as well as C21 and C22. The corresponding results indicate a reduction in off-diagonal responses, implying that FCD-DETR alleviates inter-class misclassification to a certain degree.

Nonetheless, the results demonstrate that recognition remains challenging for certain groups. The diagonal value of C14 diminishes from 0.62 to 0.56, and the accurate recognition rates of C13 likewise decrease. These instances may pertain to inter-class resemblance in wing morphology, body pigmentation, and texture, in addition to position variation, occlusion, body fragmentation, and indistinct boundaries in UV insect-trap imagery.

The comparison demonstrates that FCD-DETR enhances recognition across most categories and diminishes various inter-class confusions. Nonetheless, low-frequency, weakly textured, and morphologically similar categories remain challenging for UV insect trap pest detection.

3.9. External Generalization Evaluation on the Pest24 Dataset

To further evaluate the external generalization ability of FCD-DETR, we conducted additional experiments on the public Pest24 dataset. The results are shown in Table 10.

Table 10. External validation comparison on the Pest24 dataset.

Model	Precision	Recall	mAP@0.5	mAP@50:95	F1	FPS	Params
RT-DETR-R18	71.41	63.24	64.84	40.97	64.04	277.31	19.90
FCD-DETR	68.92	67.20	66.33	42.12	66.04	225.28	20.79

As shown in Table 10, compared with RT-DETR-R18, FCD-DETR improves *Recall* from 63.24% to 67.20%, *mAP@0.5* from 64.84% to 66.33%, *mAP@50:95* from 40.97% to 42.12%, and *F1*-score from 64.04% to 66.04%. Although *Precision* decreases slightly, the improved *Recall*, *mAP*, and *F1*-score indicate that FCD-DETR achieves better overall detection performance on the external Pest24 dataset. In addition, FCD-DETR still maintains a real-time inference speed of 225.28 FPS with only a slight increase in parameters, demonstrating its practical applicability for pest detection. To further illustrate the detection differences between the baseline and the proposed model, representative qualitative results are shown in Figure 12.

As shown in Figure 12, in representative Pest24 images, the baseline RT-DETR-R18 suffers from missed detections and false detections, whereas FCD-DETR detects pest targets more completely and accurately. This visual comparison further indicates that the proposed model improves foreground discrimination and detection robustness under external dataset conditions.

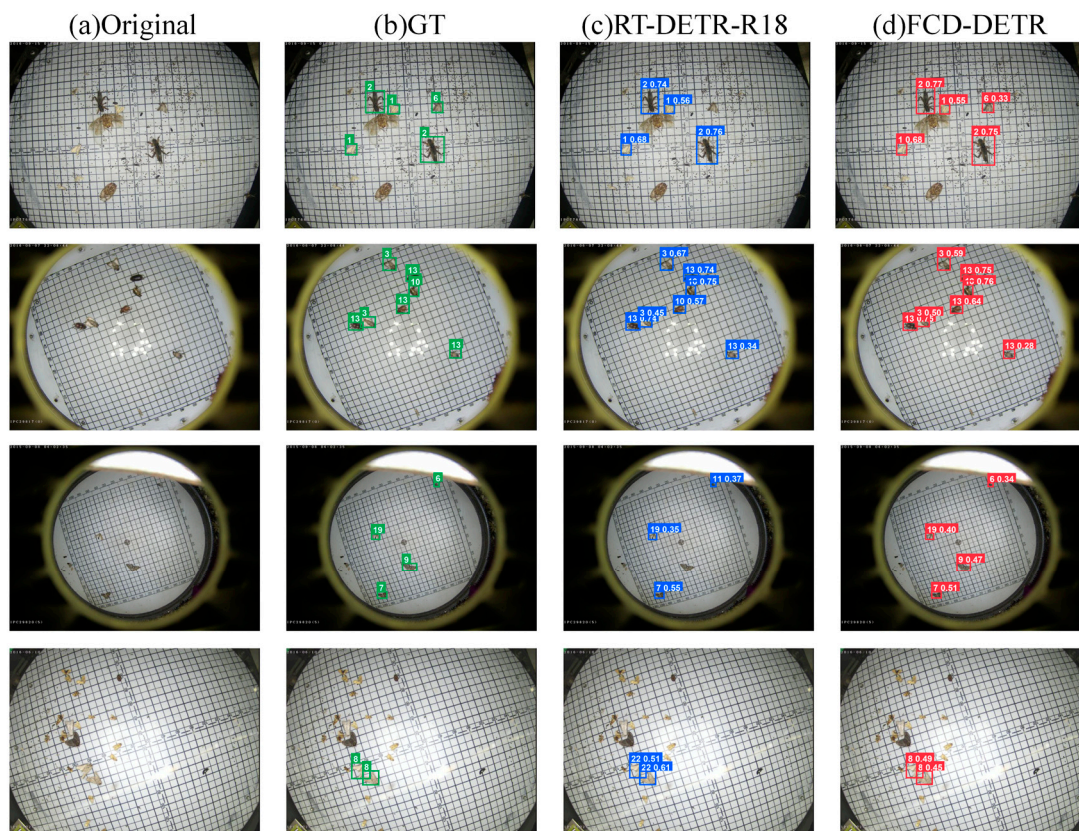


Figure 12. Qualitative comparison of detection results on the Pest24 dataset.

4. Discussion

The findings indicate that FCD-DETR enhances pest recognition in complex UV insect-trap images, implying that this task transcends a mere small-object detection issue. It encompasses the collective issues of scale variation, dense target distributions, background interference, inter-class similarity, and long-tailed category distributions. The enhancements suggest that foreground-aware multi-scale fusion, context-enhanced feature extraction, and semantic interaction are effective for improving pest foreground discrimination in these situations.

Nonetheless, several limitations remain. The dataset was primarily gathered in Meishan City, Sichuan Province, China. Despite the images being obtained from actual field monitoring situations, the model requires validation across diverse geographic areas, seasonal variations, crop varieties, insect population dynamics, and UV trap apparatuses. Moreover, although FCD-DETR helps alleviate the adverse influence of long-tailed class distribution for some low-frequency or low-AP categories, the long-tail problem has not been fully resolved. Detection performance remains unstable for certain low-frequency or morphologically similar categories, indicating that long-tailed class distributions and fine-grained inter-class confusion continue to pose challenges in multi-class pest detection. A further drawback is that FADM provides supplementary parameters while enhancing detection performance, potentially impacting deployment on resource-constrained edge devices.

Subsequent efforts will concentrate on augmenting the dataset across various areas, seasons, agricultural fields, and monitoring apparatus to enhance data diversity and model generalization. Furthermore, dedicated long-tail learning strategies, including class-balanced sampling [11], focal loss [36], semi-supervised learning, and enhanced data augmentation, will be investigated to improve the recognition of low-frequency pest

categories. Lightweight optimization techniques, including pruning, quantization [37], and knowledge distillation [38], will be examined to facilitate real-time implementation in field pest monitoring systems.

5. Conclusions

This study developed a real-world UV insect-trap pest dataset comprising 12,426 high-resolution images and 38,003 annotated instances across 36 pest categories, and introduced FCD-DETR for multi-class pest detection in intricate light-trap monitoring contexts. FCD-DETR enhances pest target detection by using foreground-aware multi-scale fusion, lightweight context-enhanced backbone feature extraction, and DHSA-based semantic interaction, effectively addressing scale variation, background interference, occlusion, and inter-class similarity.

On the constructed dataset, FCD-DETR attained 64.42% *mAP@0.5* and 41.33% *mAP@50:95*, surpassing RT-DETR-R18 by 5.25 and 5.11 percentage points, respectively. The experimental findings validate that FCD-DETR offers an effective detection framework for UV insect-trap pest image analysis. Future endeavors will concentrate on enhancing cross-region generalization, recognizing low-frequency categories, and facilitating lightweight deployment.

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

Funding: This research was funded by the Natural Science Foundation of Sichuan Province, grant number 2025ZNSFSC0162, and the College Students' Innovation and Entrepreneurship Training Program, project number S202610626143.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request. The implementation code of FCD-DETR is publicly available at <https://github.com/ytao45278-prog/FCD-DETR.git> (accessed on 9 July 2026).

Conflicts of Interest: Author Yan Cheng is employed by the company PipeChina Northwest Branch. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UV	Ultraviolet
CNN	Convolutional Neural Network
YOLO	You Only Look Once
DETR	Detection Transformer
RT-DETR	Real-Time Detection Transformer
FADM	Foreground-Aware Decoupled Mixer
DHSA	Dynamic-Range Histogram Self-Attention
C2f	Cross Stage Partial with Two Convolutions
CFBlock	Context Fusion Block
MLP	Multilayer Perceptron
FFN	Feed-Forward Network
GELU	Gaussian Error Linear Unit
AP	Average Precision

mAP	mean Average Precision
IoU	Intersection over Union
TP	True Positive
FP	False Positive
FN	False Negative
F1	F1-score
Params	Parameters

References

1. Kariyanna, B.; Sowjanya, M. Unravelling the use of artificial intelligence in management of insect pests. *Smart Agric. Technol.* **2024**, *8*, 100517. [\[CrossRef\]](#)
2. Badgujar, C.M.; Poullose, A.; Gan, H. Agricultural object detection with You Only Look Once (YOLO) algorithm: A bibliometric and systematic literature review. *Comput. Electron. Agric.* **2024**, *223*, 109090. [\[CrossRef\]](#)
3. Zhang, W.; Huang, H.; Sun, Y.; Wu, X. AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning. *Front. Plant Sci.* **2022**, *13*, 1079384. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Chen, X.; Yang, X.; Hu, H.; Li, T.; Zhou, Z.; Li, W. DAMI-YOLOv8l: A multi-scale detection framework for light-trapping insect pest monitoring. *Ecol. Inform.* **2025**, *86*, 103067. [\[CrossRef\]](#)
5. Tang, Y.; Wang, J.; Sheng, W.; Bian, J. EP-YOLO: An Enhanced Lightweight Model for Micro-Pest Detection in Agricultural Light-Trap Environments. *Sensors* **2026**, *26*, 2607. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Wang, N.; Fu, S.; Rao, Q.; Zhang, G.; Ding, M. Insect-YOLO: A new method of crop insect detection. *Comput. Electron. Agric.* **2025**, *232*, 110085. [\[CrossRef\]](#)
7. Hou, S.; Pang, Y.; Wang, J.; Hou, J.; Wang, B. RS-YOLO: A highly accurate real-time detection model for small-target pest. *Smart Agric. Technol.* **2025**, *12*, 101212. [\[CrossRef\]](#)
8. Wang, C.; Dai, Y.; Sun, P.; Zhang, L.; Wang, B.; Duan, L.; Wang, J. MRLCM-YOLO: A lightweight and multi-scale enhanced model for detecting cowpea pests in complex field environments. *Smart Agric. Technol.* **2026**, *13*, 101676. [\[CrossRef\]](#)
9. Su, B.; Zhu, Y.; Lin, Y. Pest-YOLOv8: Enhanced detection for small and complex agricultural pests using triple attention and Wise-IoU. *J. Plant Dis. Prot.* **2025**, *132*, 135. [\[CrossRef\]](#)
10. Chen, H.; Wen, C.; Zhang, L.; Ma, Z.; Liu, T.; Wang, G.; Yu, H.; Yang, C.; Yuan, X.; Ren, J. Pest-PVT: A model for multi-class and dense pest detection and counting in field-scale environments. *Comput. Electron. Agric.* **2025**, *230*, 109864. [\[CrossRef\]](#)
11. Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277. [\[CrossRef\]](#)
12. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; pp. 213–229. [\[CrossRef\]](#)
13. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
14. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
15. Liu, J.; Zhou, C.; Zhu, Y.; Yang, B.; Liu, G.; Xiong, Y. RicePest-DETR: A transformer-based model for accurately identifying small rice pest by end-to-end detection mechanism. *Comput. Electron. Agric.* **2025**, *235*, 110373. [\[CrossRef\]](#)
16. Luo, W.; Zhang, H.; Xu, L. CALDS-RTDETR: A robust forestry pest detection model for small targets in complex environments. *Comput. Electron. Agric.* **2026**, *244*, 111482. [\[CrossRef\]](#)
17. Wang, Q.-J.; Zhang, S.-Y.; Dong, S.-F.; Zhang, G.-C.; Yang, J.; Li, R.; Wang, H.-Q. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. *Comput. Electron. Agric.* **2020**, *175*, 105585. [\[CrossRef\]](#)
18. Xu, Z.; Wu, D.; Yu, C.; Chu, X.; Sang, N.; Gao, C. SCTNet: Single-branch CNN with Transformer semantic information for real-time segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024.
19. Sun, S.; Ren, W.; Gao, X.; Wang, R.; Cao, X. Restoring images in adverse weather conditions via Histogram Transformer. In *Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer: Cham, Switzerland, 2024; pp. 111–129. [\[CrossRef\]](#)
20. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

21. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 740–755. [\[CrossRef\]](#)
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning what you want to learn using programmable gradient information. In *Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer: Cham, Switzerland, 2024. [\[CrossRef\]](#)
24. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-time end-to-end object detection. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024*. [\[CrossRef\]](#)
25. Peng, Y.; Li, H.; Wu, P.; Zhang, Y.; Sun, X.; Wu, F. D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement. In *Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025*.
26. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; pp. 1580–1589. [\[CrossRef\]](#)
27. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018*; Springer: Cham, Switzerland, 2018; pp. 3–19. [\[CrossRef\]](#)
28. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; pp. 13713–13722. [\[CrossRef\]](#)
29. Zhou, S.; Chen, D.; Pan, J.; Shi, J.; Yang, J. Adapt or perish: Adaptive sparse Transformer with attentive feature refinement for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024*; pp. 2952–2963.
30. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision Transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022*; pp. 4794–4803. [\[CrossRef\]](#)
31. Pan, Z.; Cai, J.; Zhuang, B. Fast vision Transformers with HiLo attention. In *Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022*.
32. Meng, W.; Luo, Y.; Li, X.; Jiang, D.; Zhang, Z. PolaFormer: Polarity-aware linear attention for vision Transformers. In *Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025*.
33. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; pp. 10781–10790. [\[CrossRef\]](#)
34. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; pp. 821–830.
35. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; pp. 618–626. [\[CrossRef\]](#)
36. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; pp. 2980–2988. [\[CrossRef\]](#)
37. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *Proceedings of the International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016*.
38. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.

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