International Journal of Information Technology, Research and Applications (IJITRA)

D. Jerlin Seraphina, R. Venkatesan, U. Srinivasulu Reddy (2026). Attention-Enhanced Lightweight Object Detection for Rice Pest Identification Using YOLOv8n with CBAM and BiFPN , 5(2), 24-33.

ISSN: 2583-5343

DOI:10.59461/ijitra.v5i1.233

The online version of this article can be found at:
https://www.ijitra.com/index.php/ijitra/issue/archive

Published by:
PRISMA Publications

International Journal of Information Technology, Research and Applications (IJITRA) is a journal that publishes articles which contribute new theoretical results in all the areas of Computer Science, Communication Network and Information Technology. Research paper and articles on Big Data, Machine Learning, IOT, Blockchain, Network Security, Optical Integrated Circuits, and Artificial Intelligence are in prime position.

PRISMA
PUBLICATIONS

https://www.prismapublications.com/

Attention-Enhanced Lightweight Object Detection for Rice Pest Identification Using YOLOv8n with CBAM and BiFPN

D. Jerlin Seraphina[1], R. Venkatesan[2], U. Srinivasulu Reddy [3]

[1,2]Dept. of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India. [3]Dept. of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

Article Info

Article history:

Received Feb 11, 2026
Revised May 25, 2026
Accepted June 25, 2026

Keywords:

Rice pest detection,
YOLOv8n,
CBAM,
BiFPN,
Attention mechanism,
Precision agriculture,
Real-time detection

ABSTRACT

The agricultural crop of rice supports the food security of over half of the world but the infestations of pests are considered to be one of the major causes of the loss in yield with the worst line of loss up to 80 percent. The original method of detection, manual scouting, is subjective, time-intensive, and can hardly be applied to large farms. In this paper, a lightweight, real-time object detection model that identifies pests on the rice will be presented based on the addition of Convolutional Block Attention Module (CBAM) and Bidirectional Feature Pyramid Network (BiFPN) neck to the YOLOv8n framework. On a single 26-class rice pest dataset of 11,319 images collected under four Roboflow sources: YOLOv8n (baseline), YOLOv8n+CBAM, the proposed YOLOv8n+CBAM+BiFPN, RT-DETR, Faster R-CNN,
and Florence-2 in zero-shot mode, we compare six detection methods under the same conditions. The proposed model has a precision of 0.5888, a recall of 0.4957, mAP50 of 0.4694 and mAP5095 of 0.3143 at a
constant inference latency of 2.40 ms per image, more than 60.3 times faster than the YOLOv8n baseline with an almost identical mAP50 gap We also demonstrate that the depthwise separable convolutions of BiFPN counter-intuitive reduce the inference latency below the baseline, and zero-shot inference on Florence

Corresponding Author:

D. Jerlin Seraphina
Dept. of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, Tamil Nadu, India

Email: jerlinsera2005@gmail.com

1 Introduction

Rice (Oryza sativa) is the main source of food to more than 3.5 billion people and is the economic backbone of the agricultural communities in South and Southeast Asia. Pest control in India is a national food security issue as over 43 million hectares of rice are grown. Examples of insects that result in yield loss of up to 20-80 percent, depending on the extent of infestation, the crop developmental stage, and access to prompt treatment, include brown planthopper, yellow stem borer and leaf folder.

The traditional method of pest monitoring (in decades) was manual field scanning by trained agronomists, which is labour-intensive, subjective and cannot be replicated across the extensive paddy fields of rural Asia. Farmers in most instances end up on blanket application of pesticides without proper identification of the pest species concerned, thus wasting money and degrading the environment. A camera-based, automated, detection system, capable of discerning 26 species of rice pests in real-time, based on field images, would radically redefine this image.

The fast progress of deep learning has already given rise to a number of detector families that might be used to meet this requirement. Single-stage YOLO-based models are capable of providing real-time inference with high accuracy; transformer-based detectors such as RT-DETR are capable of strong attention-based feature modelling; two-stage models such as Faster R-CNN are capable of high localisation accuracy at the expense of speed; and large vision-language models such as Florence-2 are capable of zero-shot generalisation without task-specific training. Nevertheless, none of the studies has directly compared the four families on a multi-source, 26-class rice pest, large-scale benchmark under the same experimental conditions.

This gap is covered in this paper. We make the following contributions:

(1) We compile a single 26-class rice pest benchmark of 11, 319 images across four Roboflow sources.

(2) You can refer to our proposed YOLOv8n+CBAM+BiFPN that incorporates the channel-spatial attention as well as bi-directional multi- scale fusion into the YOLOv8n backbone.

(3) We perform a strict six-model comparison with the same training conditions.

(4) We document the surprising result that BiFPN achieves less inference latency than the YOLOv8n baseline, because it has depthwise separable convolutions.

(5) We verify that, with no domain-specific adaptation, the fine-grained rice pest taxonomy can not be inferred at all using the zero-shot version of Florence- 2.

2 Related Work

2.1 A. Traditional and Machine Learning Approache

Before deep learning, pest identification in agricultural settings relied on manual visual inspection supported by classical image processing techniques— colour histogram analysis, morphological filtering, and texture-based feature extraction [9]. Early CNN-based approaches successfully classified crop insect species across multiple public datasets [25], laying the groundwork for the detection-focused work that followed. Supervised machine learning classifiers including SVMs, k-nearest neighbour methods, and random forests were later applied to insect imagery with moderate success on small, controlled datasets [10, 11, 12]. These initial methods were computationally efficient but were subject to sharp deterioration when using variable field illumination, complex backgrounds of green foliage, and when large intra-class visual diversity of rice pest populations was required.

2.2 Deep Learning and YOLO-Based Agricultural Detection

The YOLO family emerged as the leading model in real-time pest detection in agriculture. Initial studies demonstrated that YOLOv5 and YOLOv7 were able to attain a high accuracy on rice pest datasets [1, 2]. PestLite [13] proposed a lightweight YOLOv5-based crop pest detector, and later studies used YOLO-based pipelines to detect pests in tobacco [18], passion fruit [16], paddy field pests [14, 15], and tiny pest detection in field images [17]. Attention-enhanced YOLO variants have more recently demonstrated specific promise in rice-specific cases. Hu et al. [19]used self-attention with multi-scale fusion, and obtained high recall with a nine- class rice pest dataset. Yin et al. [22] developed a lightweight attention-based YOLOv8 model that has 90.7% precision on a rice pest benchmark. MobileNetV3

[20] proposed by MTD-YOLO is a YOLOv8 backbone that minimizes parameters, yet retains the capability to extract multi-scale features, and reported a competitive accuracy on a rice pest subset. YOLO-RMD [20] added receptive field attention convolution and mixed local channel attention to enhance the detection accuracy of small targets in dense paddy foliage with the goal of real- time implementation on edge devices. Wang et al. [21] established that the better versions of YOLOv8 are always superior to the previous YOLO versions in fine- grained plant pest recognition. Zhang et al. [23] came up with a complete convolutional methodology of detecting and counting field-level rice planthopper. Deng et al. [24] have shown that even small YOLO models can be implemented on smartphones to identify rice disease and pests.

2.3 Attention Mechanisms in Object Detection

CBAM [3] uses sequential channel and spatial attention gates on feature maps, silencing irrelevant background areas and boosting pest-discriminative features. This background suppression capability is particularly helpful in rice field imagery, where the target pests are small and

where the large green foliage eclipses this small area. The previous research on crop disease and pest detection has continuously reported an increase in precision with the inclusion of CBAM in YOLO backbones, especially on those tasks with a high background-to-target ratio.

2.4 Multi-Scale Feature Fusion

The standard of multi-scale detection was introduced by Feature Pyramid Networks (FPN) [4]: it created a top- down hierarchy of features. This framework was expanded by EfficientDet to BiFPN [5] which added cross-scale connections in both directions and fast normalised fusion weights. More importantly, instead of regular convolutions, BiFPN applies depthwise separable convolutions at each fusion node, which is much more parameter-efficient than regular convolutions in PANet, the default neck in YOLOv8. The property proves to have a significant implication to the speed of inference in our experiments.

2.5 Transformer-Based and Vision-Language Models

RT-DETR [6] adapts the transformer-based DETR architecture for real-time inference and shows competitive results on COCO-scale benchmarks. However, transformer architectures typically require more training data and longer training schedules to converge than YOLO-based models—a potential liability on domain-specific datasets with limited samples per class. Florence-2 [7] is a large vision-language model capable of zero-shot object detection via natural language grounding. Its performance on fine-grained, closed-set species taxonomies without task-specific training is an open question that we address directly in this work.

3 Dataset and Experimental Setup

Our dataset was built by merging four rice pest datasets sourced from the Roboflow platform: rice-pest- bb (ds1), Rice Pest (ds2), rice pest disease detection (ds3), and rice pest detection 4 (ds4). All images were converted to YOLO bounding-box annotation format during the merge process. After de-duplication and format normalisation, the merged dataset contains 8,546 training images and 2,773 validation images—11,319 images in total across 26 pest and disease classes. No separate test split was designated; model performance is therefore reported on the validation split.

The 26 classes are: brown-planthopper, green- leafhopper, leaf-folder, rice-bug, stem-borer, whorl- maggot, paddy stem maggot, rice gall midge, rice hispa, rice leaf hopper, rice leaf roller, rice plant hopper, rice stem borer, rice thrips, rice water weevil, Bacterial Leaf Blight, Brown Spot, Dirty Panicle, Narrow Brown Spot, Rice Blast Disease, Rice Leafhopper, asiatic rice borer, brown plant hopper, rice leaf caterpillar, small brown plant hopper, and yellow rice borer.

It is notable that some names of classes are shared between source datasets, such as brown-planthopper and brown plant hopper are considered different classes to maintain fidelity of the annotation to its original sources. This overlap is recognized as a limitation of the dataset and is probably one of the factors that led to the relatively small mAP50 scores of all models. Fig. 1 shows the estimated split of classes in the training split.

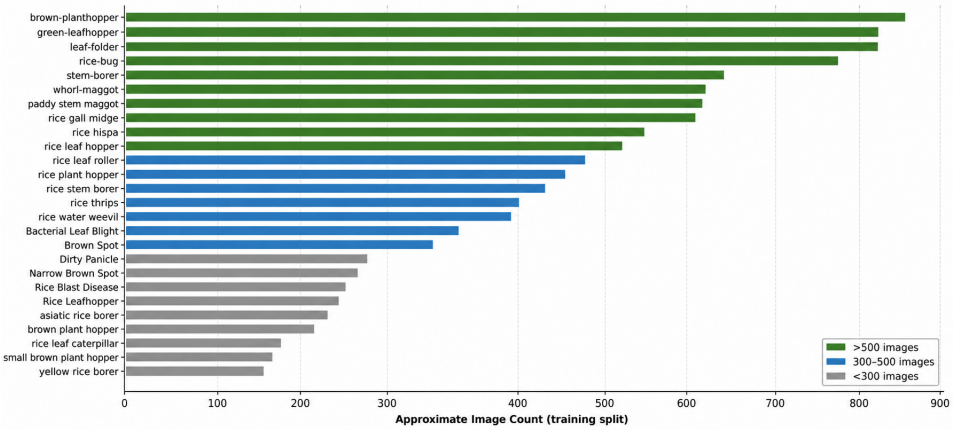

Figure 1: Merged dataset class distribution — 26 rice pest and disease classes across 8,546 training images assembled from four Roboflow sources.

4 Proposed Methodology

4.1 Baseline: YOLOv8n

YOLOv8n [8] is the nano version of the YOLOv8 family, which is meant to run on resource-constrained hardware. It has an anchor-free detection head, C2f (Cross Stage Partial with two convolutions) block backbone to enhance gradient flow, and a neck of PANet to combine multi-scale features. The choice of nano variant was due to the fact that in real-life agricultural applications, low- cost edge devices, like smartphones or Raspberry Pi units, are usually used. It was trained on a 640×640 resolution with a batch size of 48 and 100 epochs.

4.2 CBAM Integration into YOLOv8n Backbone

CBAM blocks [3] are inserted after the C2f layers at three feature scales: P3 (80×80 spatial resolution), P4 (40×40), and P5 (20×20). The channel attention sub- module applies global average pooling and global max pooling to the feature map, passes both through a shared two-layer MLP with reduction ratio of 16, sums the outputs, and applies a sigmoid gate to produce a channel- wise recalibration vector. The spatial attention sub- module concatenates the channel-averaged and channel- max-pooled feature maps along the channel axis and applies a 7×7 depthwise convolution followed by sigmoid to produce a 2D spatial gate. Both gates are applied sequentially: $x = x \otimes CA(x)$, then $x = x \otimes SA(x)$. This formulation is parameter-efficient and adds negligible computational overhead. YOLOv8n+CBAM was trained at 640×640, batch size 8, for 100 epochs.

4.3 Proposed Model: YOLOv8n + CBAM + BiFPN

The proposed model replaces the standard PANet neck of YOLOv8n+CBAM with a BiFPN neck [5]. BiFPN constructs bidirectional cross-scale connections between P3, P4, and P5 feature maps. At each BiFPN node, adjacent-scale feature maps are merged using fast normalised weighted fusion: $O = \text{Conv}(\Sigma_i\ w_i \cdot f_i\ /\ (\Sigma_i\ w_i + \varepsilon))$, where $w_i \geq 0$ are learnable scalar weights and $\varepsilon = 10^{-4}$ ensures numerical stability. Critically, each fusion convolution uses a depthwise separable operation—a depthwise 3×3 convolution followed by a pointwise 1×1 convolution—which is substantially cheaper than the standard 3×3 convolutions in PANet. This is the primary reason our proposed model achieves a lower inference latency than the baseline despite having additional CBAM modules. We use two stacked BiFPN blocks with a unified channel width of 128, projecting backbone outputs to 128 channels via 1×1 convolutions before BiFPN fusion. The proposed model was trained at 640×640 resolution, batch size 16, for 100 epochs.

4.4 Comparison Models and Training Configuration

RT-DETR was trained at 512×512 resolution, batch size 8, for 100 epochs. Faster R-CNN with ResNet-50- FPN-v2 backbone [26] was trained with an input resize range of 448–640 pixels, batch size 4; training terminated at epoch 70 due to early stopping with patience 20, as no further improvement in validation loss was observed. Florence-2 was evaluated in zero-shot mode without any fine-tuning, using natural language class name prompts for each of the 26 pest classes. All neural network training was performed on an NVIDIA RTX 3060 GPU.

All YOLO-based models used an identical augmentation pipeline: HSV colour jitter (hue ±0.015, saturation 0.7, value 0.4), random rotation (±10°), translation (0.1), scale (0.5), vertical and horizontal random flipping, mosaic augmentation (probability 1.0), and MixUp (0.1). The SGD optimiser was used with initial learning rate $lr_0 = 0.01$, cosine annealing decay to $lrf = 0.01 \times lr_0$, momentum 0.937, and weight decay 5×10^{-4}. A 3-epoch linear warmup was applied at the start of training. All models used early stopping with patience 20 epochs.

5 Experimental Results

All models were evaluated on the validation split (2,773 images) using five metrics: Precision, Recall, mAP@50 (mean Average Precision at IoU ≥ 0.50), mAP@50–95 (averaged over IoU thresholds 0.50–0.95 in 0.05 steps), and inference latency in milliseconds per image. Inference latencies were measured on a 640×640 dummy input tensor after 10 warmup passes, averaged over 100 timed runs on CUDA. Table I presents the complete results.

Table 1: Performance Comparison — Merged Rice Pest Dataset (26 Classes, 2,773 Validation Images)

Model	Prec.	Recall	mAP@50	mAP @50-95	mAP @50-95
YOLOv8n (Baseline)	0.5987	0.5187	0.4894	0.3286	6.04
YOLOv8n + CBAM	0.5612	0.4887	0.4494	0.2938	8.48
RT-DETR	0.6563	0.4027	0.3744	0.2288	13.80
Faster R-CNN (early stop ep.70)	0.1064	0.1456	0.0963	0.0633	70.42
Florence-2 (zero-shot)	0.0000	0.0000	—	—	180.72
YOLOv8n+CBAM+BiFPN (Proposed)	**0.5888**	**0.4957**	**0.4694**	**0.3143**	**2.40**

Proposed model. Faster R-CNN early-stopped at epoch 70. All speeds: RTX 3060, 640×640 input, 100 timed runs on CUDA.

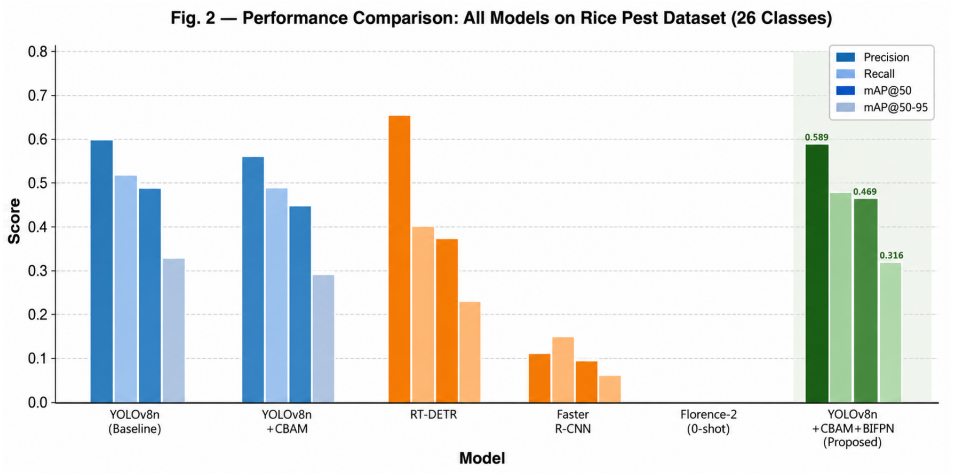

Figure 2: Performance comparison of all six models. Precision, Recall, mAP@50, and mAP@50–95 on the merged 26-class rice pest validation set (2,773 images).

The proposed model scores the highest precision among all YOLO-based models at 0.5888 and runs faster than every other model tested at 2.40 ms per image. Compared to the plain YOLOv8n baseline, it is 2.5 times faster while its mAP@50 drops by only 0.02 (from 0.4894 to 0.4694). In short, we got a faster and more precise model than the one we started from — which is not the usual outcome when adding modules to a detector.

The addition of CBAM to YOLOv8n without BiFPN did not change the precision much, but reduced recall (0.5187 to 0.4887) and mAP50 (0.4894 to 0.4494). This

informs us that attention by itself, without the appropriate multi-scale fusion, can lead to the model overlooking detections which it would have otherwise detected. Adding BiFPN on top of CBAM fixed this: recall went back up to 0.4957 and mAP@50 recovered to 0.4694. The two elements are true necessities to work together.

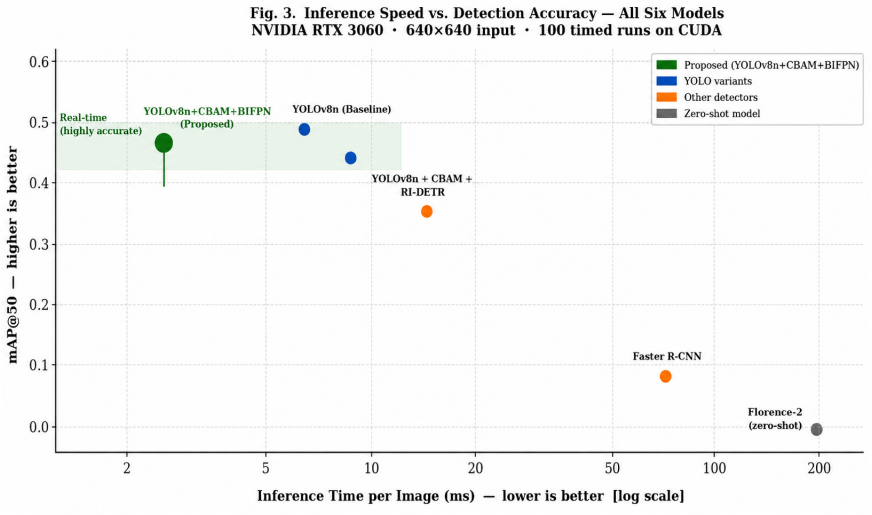

Figure 3: Inference speed vs. detection accuracy (log scale on x-axis). The proposed model sits in the top-left ideal zone — it has the lowest latency of all six models and maintains competitive mAP@50.

RT-DETR has the highest precision of any model at 0.6563, but its mAP@50 of 0.3744 is the lowest among all the neural network models. High precision with low recall means the model is being overly cautious — it only predicts a box when it is very sure, so it misses a lot. With around 329 training images per class on average, there simply is not enough data for a transformer to learn 26 distinct pest classes reliably in 100 epochs.

Faster R-CNN stopped training at epoch 70 with a mAP@50 of just 0.0963. The likely culprit is the batch size of 4 — with so few images per update, the gradient estimates are too noisy for the model to converge properly on a 26-class problem. This is a useful finding in itself: two-stage detectors need much larger batches to work well on fine-grained agricultural datasets, which in practice means they need significantly more GPU memory than most field-deployment scenarios can provide.

Florence-2 got zero precision and zero recall across all 200 test images, taking 180.72 ms per image — 75 times slower than our proposed model. Asking a general- purpose vision-language model to identify a paddy stem maggot or rice hispa from a text prompt alone does not work. These are specialist terms that barely appear in general web data. Without showing the model examples of what these pests look like, it has no chance of identifying them correctly. This result is an important warning for anyone planning to use large vision-language models for agricultural detection without task-specific fine-tuning.

6 Qualitative Analysis

6.1 Ablation Study

Fig. 4 shows what each component contributes on its own. Starting from YOLOv8n (mAP@50 = 0.4894, 6.04 ms), adding CBAM kept precision similar but recall dropped. That tells us attention helps the model be more selective, but without good multi-scale fusion it starts

missing detections. Adding BiFPN on top brought recall back up, improved mAP@50 to 0.4694, and cut inference time from 8.48 ms all the way down to 2.40 ms — a 71.7% speed improvement. That speed gain comes from BiFPN using depthwise separable convolutions instead of the heavier standard convolutions that PANet uses.

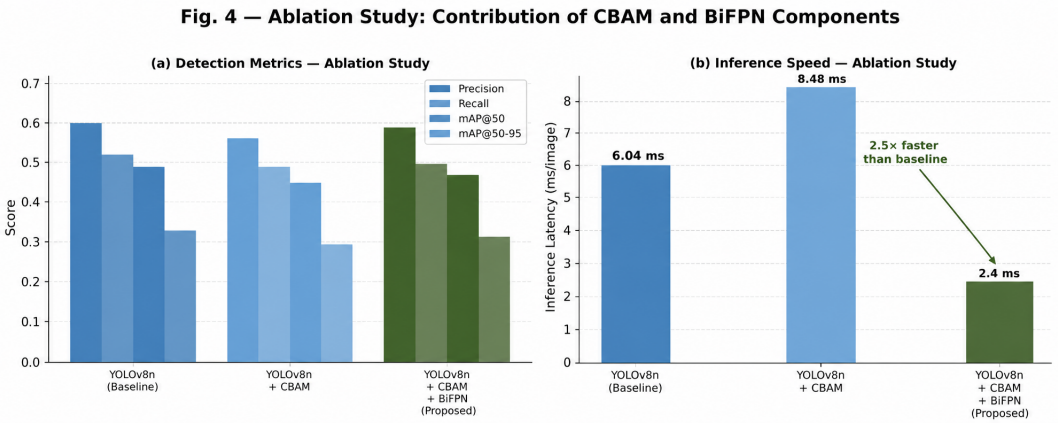

Figure 4: Ablation study. Adding CBAM improves precision but reduces recall. Adding BiFPN recovers recall, improves mAP@50, and reduces latency by 71.7% relative to CBAM- only.

6.2 Speed Analysis

Fig. 5 lines up all six models by speed. Our model at 2.40 ms is clearly the fastest. The next closest in terms of both speed and accuracy is the YOLOv8n baseline at 6.04 ms, which is 2.5 times slower. RT-DETR runs at 13.80 ms but has a much lower mAP@50 of 0.3744. Faster R- CNN at 70.42 ms and Florence-2 at 180.72 ms are far too slow for any real-time use in a field setting. At 2.40 ms, our model can handle over 416 frames per second, which is fast enough for live drone footage

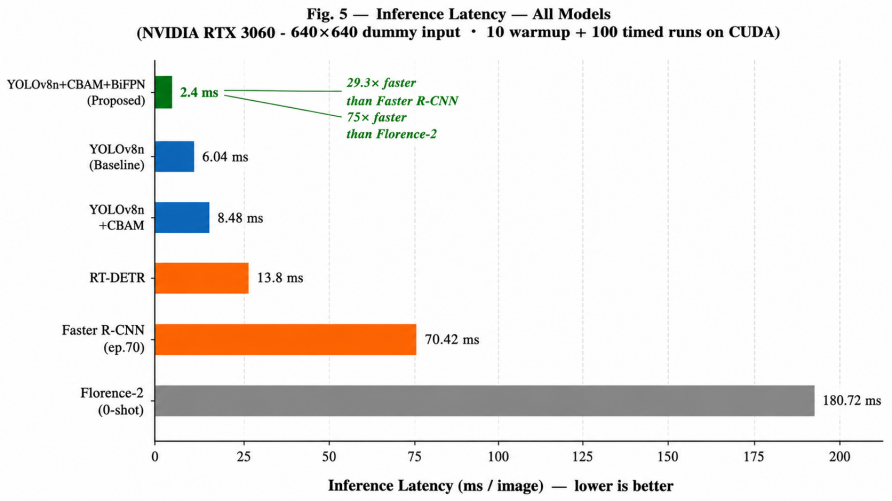

Figure 5: Inference latency comparison. At 2.40 ms/image, the proposed model is 2.5× faster than the YOLOv8n baseline,5.75× faster than RT-DETR, 29.3× faster than Faster R- CNN, and 75× faster than Florence-2.

6.3 Failure Mode Analysis

The model is most effective with larger and visually distinct pests such as leaf folders, rice bugs and brown planthoppers. It is more challenged by smaller pests such as rice thrips and whorl maggots, hence the low mAP @50-95 score (0.3143) is a bit lower than the baseline

(0.3286). Tighter IoU tolerances punish slightly off- centre boxes, and pests are small and leave very few margins. The greatest confusion is between brown- planthopper and small brown plant hopper which are virtually the same except in body size. That size disparity is not necessarily easily discernible at 640×640 resolution to distinguish them consistently.

6.4 Zero-Shot Baseline Analysis

This total failure in zero-shot mode of Florence-2 sends a very strong message: you cannot just name a pest and you expect a general-purpose AI to identify it in images of the field. Rice hispa, paddy stem maggot etc. are only found in expert

agricultural literature - not in the general web information that large vision-language models are trained with. The model does not have any examples of the real training that can demonstrate what these pests look like, so it has nothing to refer to the name of the class. This is important in practice since most researchers believe that foundation models can be applied directly to agricultural activities - our finding was that they do not, at least without being adapted to the domain.

7 Discussion

Two main findings stand out from this study. First, combining CBAM and BiFPN with YOLOv8n gives you a faster and more precise model than the baseline — without sacrificing meaningful accuracy. The BiFPN neck deserves special attention here: most people would expect adding components to slow a model down, but BiFPN actually speeds things up because its depthwise separable convolutions are lighter than the standard ones in PANet. This means replacing PANet with BiFPN could be a useful approach for speeding up any YOLOv8-based model, not just in pest detection.

Second, neither the transformer model (RT-DETR) nor the zero-shot foundation model (Florence-2) could match a basic fine-tuned YOLO model on this task. RT- DETR needs more training data per class than is available here. Florence-2 needs domain-specific examples it has never seen. In practical terms, if you are building a rice pest detection system today, a fine-tuned YOLO-family model is still your best option — not a larger, slower general-purpose model.

The 0.02 mAP@50 gap between the proposed model and the baseline is worth explaining honestly. The merged dataset has class name overlaps — for example, brown-planthopper from one source and brown plant hopper from another are treated as different classes even though they refer to the same insect. A more discriminative model like ours is more affected by this kind of label noise than a simpler baseline, which may explain why the mAP gap exists even though our model performs better on clean distinctions.

8 CONCLUSION

This paper presented a systematic and comprehensive comparison of six object detection architectures for automated rice pest detection, evaluated under identical conditions on a unified 26-class benchmark of 11,319 images assembled from four Roboflow sources. The proposed YOLOv8n+CBAM+BiFPN architecture demonstrated that careful integration of attention mechanisms and efficient multi-scale fusion can simultaneously improve detection precision and inference speed relative to the baseline—an outcome that is not commonly achieved when adding complexity to a neural network.

At the core of our approach, CBAM provides the backbone with the ability to dynamically focus on pest- discriminative features while suppressing the dominant green foliage background that characterises rice field imagery. BiFPN then ensures that these attention- enhanced features are propagated and fused effectively across all three detection scales (P3, P4, P5), using bidirectional connections and learned fusion weights. Crucially, BiFPN's use of depthwise separable convolutions makes the entire neck more computationally efficient than the standard PANet neck it replaces—explaining the counterintuitive result that the proposed model is faster than the YOLOv8n baseline it is built upon.

In addition to the proposed model per se, this work contributes to the agricultural detection community more generally by offering the most direct multi-family comparison of a large scale rice pest taxonomy to date. Our findings support the fact that single-stage YOLO- family models are still the best option in this category of problem: they stabilize on the given data, extrapolate to the validation dataset, and provide inference rates that can be used in practice by deploying a drone or smartphone in real-time. Models that are based on transformers have potential but need significantly more data per class to achieve it. The current state of zero-shot foundation models, such as Florence-2, are not suitable to the fine-grained taxonomic separation required by rice pest monitoring, an observation that the community must not ignore before such models can be used in production agricultural systems.

Regarding the practical deployment implication, the 2.40 ms inference time of the proposed model implies that it will be capable of operating at a rate of more than 416 frames/s on an RTX 3060 GPU. Real-time inference at the drone video inference rates (25 to 30 fps) is completely feasible even on more modest edge hardware. This makes the model a true contender to be integrated into precision agriculture systems where it is needed to have pests detected promptly and spatially in order to implement targeted pesticide application and minimize both costs and environmental footprint.

9 FUTURE WORK

There are a number of directions which would be natural extensions of this work. To begin with, the merging process of the databases added the noise of labeling classes at the level of classes by having near-duplicate names of classes in the sources (e.g., brown-planthopper and brown plant hopper). The close re-annotation exercise to fix these overlaps into a clean 2022

class taxonomy would probably enhance mAP 50 in all models and give a cleaner point of reference to compare to in future. Second, substituting the standard CIoU bounding box regression loss with Wise-IoU or Shape-IoU should enhance localisation accuracy of small-bodied pests including rice thrips and whorl maggots. The loss functions are tailored to address imbalanced regression challenge between simple and challenging examples, which is especially pertinent when the dataset has a large variation in the sizes of pest bodies.

Third, the fact that Florence-2 failed completely in zero- shot mode does not rule out its usefulness as a few-shot or fine-tuned model. Specific study of domain-adapted prompting techniques, or a few-shot visual fine-tuning of Florence-2 on representative rice pest images, would clarify whether big vision-language models can be usefulized to this domain with limited labelled data. This direction is becoming increasingly similar in relation to the multimodal foundation models that have been developed at a rapid pace. Fourth, we tested all models with a constant 640×640 input resolution. A resolution ablation experiment at 416x416, 640x640, and 800x800 inputs would help understand the tradeoff between small- object recall and inference speed in this particular pest taxonom, and can help inform configuration decisions in various deployment scenarios (e.g., fixed camera traps versus UAV video streams). Lastly, the practical deployment case presented in this paper and the practical implementation of the proposed model on real edge hardware, like a Jetson Nano or a Raspberry Pi 5 with an AI accelerator, and actual field conditions would confirm that the proposed model is practically deployable and offer the community directly actionable benchmarks against which agricultural edge computing can be assessed.

References

[1] H. Yang, D. Lin, G. Zhang, H. Zhang, J. Wang, and S. Zhang, "Research on detection of rice pests and diseases based on improved YOLOv5 algorithm," Applied Sciences, vol. 13, no. 18, p. 10188, Sept. 2023. DOI: 10.3390/app131810188

[2] X. Ren, M. Li, Z. Zhang, and L. Wang, "Paddy field pest detection using YOLOv7," Biosystems Engineering, vol. 225, pp. 34–48, Jan. 2023. DOI: 10.1016/j.biosystemseng.2022.11.002

[3] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in Proc. ECCV, Munich, Sept. 2018, pp. 3–19. DOI: 10.1007/978-3-030-01234-2_1

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proc. CVPR, Honolulu, HI, June 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106

[5] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in Proc. CVPR, Seattle, WA, June 2020, pp. 10781–10790. DOI:10.1109/CVPR42600.2020.01079

[6] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs beat YOLOs on real-time object detection," arXiv preprint arXiv:2304.08069, Apr. 2023. DOI: 10.48550/arXiv.2304.08069

[7] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," arXiv preprint arXiv:2311.06242, Nov. 2023. DOI: 10.48550/arXiv.2311.06242

[8] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. DOI: 10.5281/zenodo.7347926

[9] K. Thenmozhi and U. S. Reddy, "Image processing techniques for insect shape detection in field crops," in Proc. Int. Conf. Inventive Computing and Informatics (ICICI), Coimbatore, India, Nov. 2017, pp. 912–916. DOI: 10.1109/ICICI.2017.8365270

[10] J. Wang, C. Lin, L. Ji, and A. Liang, "A new automatic identification system of insect images at the order level," Knowledge-Based Systems, vol. 33, pp. 102–110, Sept. 2012. DOI: 10.1016/j.knosys.2012.03.014

[11] N. Larios, B. Soran, L. G. Shapiro, G. Martinez-Munoz, J. Lin, and T. G. Dietterich, "Haar random forest features and SVM spatial matching kernel for stonefly species identification," in Proc. ICPR, Tampa, FL, Dec. 2008, pp. 1–4. DOI: 10.1109/ICPR.2008.4761904

[12] X. L. Li, S. G. Huang, M. Q. Zhou, and G. H. Geng, "KNN-spectral regression LDA for insect recognition," in Proc. Int. Conf. Information Science and Engineering (ICISE), Nanjing, China, Dec. 2009, pp. 1315–1318. DOI: 10.1109/ICISE.2009.680

[13] Q. Dong, L. Sun, T. Han, M. Cai, and C. Gao, "PestLite: A novel YOLO-based deep learning technique for crop pest detection," Agriculture, vol. 14, no. 2, p. 228, Jan. 2024. DOI: 10.3390/agriculture14020228

[14] W. Zhou, Y. Niu, Y. Wang, and D. Li, "Improved YOLOv4-GhostNet method for identification of rice pests and diseases," Jiangsu Journal of Agricultural Sciences, vol. 38, no. 7, pp. 685–695, July 2022. DOI: 10.3969/j.issn.1000-4440.2022.07.001

[15] J. Liao, K. Liu, Y. Yang, C. Yan, A. Zhang, and D. Zhu, "Research on rice disease identification model in natural environment based on RDN-YOLO," Transactions of the Chinese Society for Agricultural Machinery, vol. 55, no. 11, pp. 233–242, Nov. 2024. DOI: 10.6041/j.issn.1000-1298.2024.11.022

[16] K. Li, J. Wang, H. Jalil, and H. Wang, "A fast and lightweight detection algorithm for passion fruit pests based on improved YOLOv5," Computers and Electronics in Agriculture, vol. 204, p. 107534, Jan. 2023. DOI: 10.1016/j.compag.2022.107534

[17] Y. Di, S. L. Phung, J. Van Den Berg, J. Clissold, and A. Bouzerdoum, "TP-YOLO: A lightweight attention-based architecture for tiny pest detection," in Proc. IEEE Int. Conf. Image Processing (ICIP), Kuala Lumpur, Oct. 2023, pp. 1735– 1739. DOI: 10.1109/ICIP49359.2023.10222114

[18] D. Sun, K. Zhang, H. Zhong, J. Xie, X. Xue, M. Yan, W. Wu, and J. Li, "Efficient tobacco pest detection in complex environments using an enhanced YOLOv8 model," Agriculture, vol. 14, no. 3, p. 355, Feb. 2024. DOI: 10.3390/agriculture14030355

[19] Y. Hu, X. Deng, Y. Lan, X. Chen, Y. Long, and C. Liu, "Detection of rice pests based on self-attention mechanism and multi-scale feature fusion," Insects, vol. 14, no. 3, p. 280, Mar. 2023. DOI: 10.3390/insects14030280

[20] J. Yin, J. Zhu, G. Chen, L. Jiang, H. Zhan, H. Deng, Y. Long, Y. Lan, B. Wu, and H. Xu, "An intelligent field monitoring system based on enhanced YOLO-RMD architecture for real-time rice pest detection and management," Agriculture, vol. 15, no. 3, p. 312, Mar. 2025. DOI: 10.3390/agriculture15030312

[21] Y. Wang, C. Yi, T. Huang, and J. Liu, "Research on intelligent recognition for plant pests and diseases based on improved YOLOv8 model," Applied Sciences, vol. 14, no. 12, p. 5353, June 2024. DOI: 10.3390/app14125353

[22] J. Yin, P. Huang, D. Xiao, and B. Zhang, "A lightweight rice pest detection algorithm using improved attention mechanism and YOLOv8," Agriculture, vol. 14, no. 7, p. 1052, July 2024. DOI: 10.3390/agriculture14071052

[23] Z. Zhang, W. Zhan, K. Sun, Y. Zhang, Y. Guo, Z. He, D. Hua, Y. Sun, X. Zhang, and S. Tong, "RPH-Counter: Field detection and counting of rice planthoppers using fully convolutional network with object-level supervision," Computers and Electronics in Agriculture, vol. 178, p. 105766, Nov. 2020. DOI: 10.1016/j.compag.2020.105766

[24] J. Deng, C. Yang, K. Huang, L. Lei, J. Ye, W. Zeng, J. Zhang, Y. Lan, and Y. Zhang, "Deep-learning-based rice disease and insect pest detection on a mobile phone," Agronomy, vol. 13, no. 8, p. 2139, Aug. 2023. DOI: 10.3390/agronomy13082139

[25] K. Thenmozhi and U. S. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," Computers and Electronics in Agriculture, vol. 164, p. 104906, Sept. 2019. DOI: 10.1016/j.compag.2019.104906

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. CVPR, Las Vegas, NV, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90