International Journal of Information Technology (IJIT) – Volume 12 Issue 6, Nov-Dec 2026

RESEARCH ARTICLE OPEN ACCESS

Deep Learning Architectures for Multi-Class Classification of Insect Pests

Narsaiah Battu[1], Arathi Chitla[2]

[1] *Research Scholar, Department of Computer Science & Engineering, Telangana University*
Dichpally, Nizamabad, Telangana, India. Email: vnbattu6@gmail.com

[2] *Professor, Department of Computer Science & Engineering, Telangana University*
Dichpally, Nizamabad, Telangana, India Email: arathichita@gmail.com

ABSTRACT

Insect pests are a major cause of crop loss, and automated image-based recognition can support timely, targeted pest control. This paper compares three deep learning architectures that embody different design principles: a convolutional network (ResNet50), a global self-attention Vision Transformer (ViT-B/16) and a hierarchical shifted-window transformer (Swin-B). All three were fine-tuned under one identical protocol on IP102, a fine-grained, long-tailed benchmark of 75,222 images in 102 pest classes and evaluated on its official 22,619-image test set. Swin-B achieved the best Top 1 accuracy (76.38%), Top 5 accuracy (91.73%) and macro F1-score (69.62%), followed by ViT-B/16 (75.71% Top 1) and ResNet50 (71.25%). ResNet50, however, produced the best-ranked and best-calibrated probabilities (macro-ROC-AUC 0.9778, micro average precision 0.7713), whereas both transformers were overconfident at high confidence. Error analysis showed that ten class pairs appear among the fifteen most frequent confusions of all three models and account for 9–10% of all errors; six of these pairs are parent and child labels in the dataset's taxonomy, such as the family Miridae and its species tarnished plant bug. Every model failed on 17.4% of test images, while at least one model was correct on 82.6%. Grad-CAM visualisations showed that ResNet50 and ViT-B/16 usually attend to the pest, whereas Swin-B heatmaps were often diffuse or fell on watermarks. The results indicate that transformers improve accuracy on fine-grained pest images, but label quality and calibration limit what any architecture can achieve.

Keywords **-** insect pest recognition, fine-grained classification, convolutional neural network, vision transformer, Swin Transformer, model calibration, Grad-CAM, precision agriculture

I. INTRODUCTION

Insect pests reduce crop yield and quality worldwide, and their control depends on correct and timely identification. Manual identification requires entomological expertise that is scarce in many farming regions, which has made automated image-based pest recognition an active application of machine learning in agriculture [1]– [3]. The task is difficult because many pest species look alike, a single species appears as egg, larva, pupa and adult, and field photographs contain cluttered backgrounds [1].

Convolutional neural networks (CNNs) such as ResNet [4] have long been the standard backbone for this task. Vision Transformers (ViT) [5] instead model global relations between image patches through self-attention, and the Swin Transformer [6] restricts attention to shifted local windows inside a hierarchical feature pyramid. These architectures make different assumptions about locality and scale, which may matter for fine-grained insect images, but they are rarely compared under one identical training protocol on a large pest benchmark.

This paper compares ResNet50, ViT-B/16 and Swin-B on IP102 [1], the largest public pest classification benchmark, using the same pre-processing, optimiser, schedule, loss and test set for all three. The contributions are:

1) a controlled comparison of a CNN, a global-attention transformer and a hierarchical window transformer on 102 fine-grained pest classes;
2) an evaluation beyond accuracy, covering macro and weighted F1, ROC-AUC, average precision and probability calibration;
3) an error analysis showing that the most frequent confusions are shared across architectures and largely reflect overlapping labels in the dataset taxonomy;
4) a model-agreement analysis that measures how many test images no architecture can classify; and
5) a Grad-CAM [7] comparison of where each architecture looks, which exposes watermarks and other dataset artefacts.

Section II reviews related work, Section III describes the data, models and protocol, Section IV reports the results, Section V discusses them and Section VI concludes.

II. Related Work

Pest recognition with CNNs. Wu *et al.* [1] introduced IP102 with 75,222 images and reported 49.4% accuracy for a fine-tuned ResNet-50, showing the difficulty of the benchmark. Later CNN studies improved this substantially: Ayan *et al.* [8] reached 67.13% with a genetic-algorithm-weighted ensemble of CNNs, Ayan [9] reached 71.84% with genetic-algorithm hyper-parameter optimisation, and Nanni *et al.* [10] reported 73.46% with an ensemble of CNNs trained with different data augmentation and optimiser variants.

Transformers for pest images. Transformers have recently produced the strongest single-model results. Wang *et al.* [11] combined a Swin Transformer with attention-augmented residual blocks (Swin-AARNet) and reported 78.77% on IP102. Such studies usually propose a new architecture; few compare standard CNN and transformer backbones under the same training recipe.

Data quality. IP102 was collected from the internet and contains label noise. Peng *et al.* [12] screened it for misplaced categories and images without a visible pest and retained only 47,393 of the 75,222 images in their cleaned HQIP102 version. Label noise and artefacts such as watermarks can encourage shortcut learning [13], a risk this paper examines with Grad-CAM.

III. Materials and Methods

A. Dataset

IP102 [1] contains 75,222 images of 102 insect pest classes affecting eight crops: rice, corn, wheat, beet and alfalfa (field crops) and vitis, citrus and mango (economic crops). The class distribution is strongly long tailed. The official split was used (Table I); all reported results are on the untouched test set.

TABLE I
Dataset Split

Subset	Images	Share (%)
Training	45,095	59.9
Validation	7,508	10.0
Test	22,619	30.1
Total (102 classes)	75,222	100.0

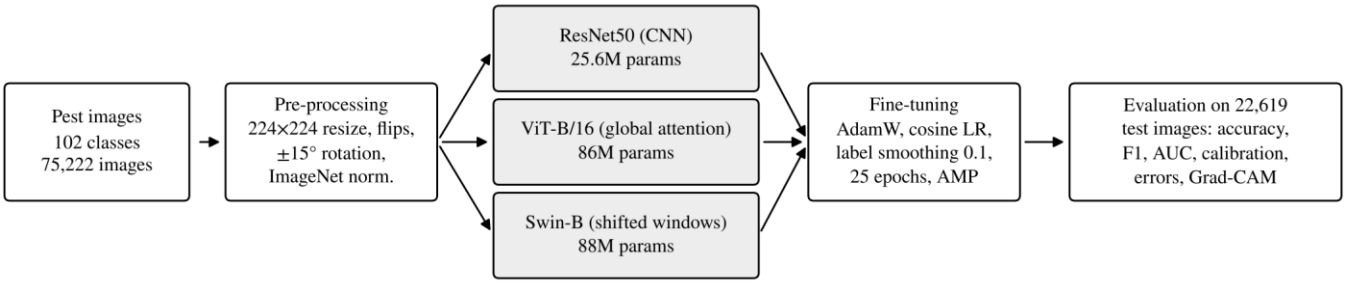


Fig. 1. Overview of the experimental pipeline. The three architectures share identical pre-processing, training and evaluation.

B. Architectures

1) ResNet50: ResNet [4] stacks bottleneck blocks of 1 × 1, 3 × 3 and 1 × 1 convolutions with identity shortcut connections, so each block learns a residual function:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{1}$$

Convolutions build in locality and translation equivariance, an inductive bias that helps with limited data.

2) ViT-B/16: The Vision Transformer [5] splits a 224 × 224 image into 196 non-overlapping 16 × 16 patches, embeds them as tokens with learnable position embeddings and a classification token, and applies 12 transformer encoder layers. Each layer uses multi-head scaled dot-product attention [14]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Every patch can attend to every other patch from the first layer, so the model has a global receptive field but little built-in locality bias, and it learns representations that differ markedly from those of CNNs [15].

3) Swin-B: The Swin Transformer [6] computes self-attention within non-overlapping M × M windows (M = 7) and shifts the window grid between consecutive layers so that information flows across windows. Patch merging halves the spatial resolution at each stage, producing a CNN-

like feature pyramid. For an h × w token map with C channels, window attention reduces the cost of global attention from quadratic to linear in the number of tokens:

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC \quad (3)$$

Table II summarises the three designs.

TABLE II
ARCHITECTURAL COMPARISON OF THE THREE MODELS

Property	ResNet50	ViT-B/16	Swin-B
Core operation	3 × 3 convolution, residual blocks	Global multi-head self-attention	Shifted-window self-attention
Feature maps	Hierarchical (4 stages)	Single scale (14 × 14 tokens)	Hierarchical (4 stages)
Receptive field	Local, grows with depth	Global from layer 1	Local windows, expanded by shifting
Inductive bias	High (locality)	Low	Moderate
Parameters	25.6M	86M	88M

Approximate parameter counts of the ImageNet models [6], [16].

C. Training Protocol

All models were created with the timm library [16] from ImageNet [17] pre-trained checkpoints, with the classifier replaced by a 102-way linear layer, and fine-tuned end to end in PyTorch [18] with the identical protocol in Table III. Cross-entropy with label smoothing [19] replaces the one-hot target with

$$q_k = (1 - \varepsilon)\,\delta_{k,y} + \frac{\varepsilon}{K} \quad (4)$$

where y is the true class, K = 102 and ε = 0.1. The checkpoint with the highest validation accuracy was used for testing.

TABLE III
COMMON TRAINING CONFIGURATION

Setting	Value
Input size	224 × 224, ImageNet mean/std normalisation
Augmentation	Horizontal flip (p = 0.5), vertical flip (p = 0.2), rotation ±15°
Optimiser	AdamW [20], learning rate 1×10^{-4}, weight decay 0.01
Schedule	Cosine annealing [21] over 25 epochs
Loss	Cross-entropy with label smoothing ε = 0.1
Batch size, precision	32; automatic mixed precision [22]
Model selection	Best validation accuracy
Hardware	Google Colab GPU runtime

D. Evaluation

On the test set we report Top 1 and Top 5 accuracy; macro- and weighted-average precision, recall and F1-score [23]; one-versus-rest macro-ROC-AUC; and micro-averaged average precision (AP). Calibration [24] was assessed with a reliability diagram over 15 confidence bins of the one-versus-rest probabilities. Errors were analysed through the fifteen most frequent off-diagonal confusion pairs of each model, and complementarity through the number of models (0–3) that classify each test image correctly. Grad-CAM [7] was computed on the last residual block of ResNet50 and on the last normalisation layer of ViT-B/16 and Swin-B, with token maps reshaped to 2-D grids.

IV. RESULTS

A. Overall Performance

Swin-B achieved the best Top 1 accuracy (76.38%), Top 5 accuracy (91.73%), macro recall and macro F1-score (Table IV). ViT-B/16 was close in Top-1 accuracy (75.71%) and had the highest macro precision, but the lowest Top-5 accuracy (90.61%). ResNet50 was 5.13 points below Swin-B in Top-1 accuracy, with 6,502 test errors against 5,343 for Swin-B.

TABLE IV
TEST-SET RESULTS OF THE THREE MODELS (22,619 IMAGES)

Model	Top-1 (%)	Top-5 (%)	Macro P (%)	Macro R (%)	Macro F1 (%)	Weighted F1 (%)	Macro AUC	Micro AP
ResNet50	71.25	90.99	65.68	62.22	63.22	70.77	**0.9778**	**0.7713**
ViT-B/16	75.71	90.61	**71.22**	67.43	68.85	75.32	0.9635	0.7650
Swin-B	**76.38**	**91.73**	70.93	**68.90**	**69.62**	**76.07**	0.9672	0.7609

P: precision; R: recall. Macro averages weight all 102 classes equally; weighted averages weight classes by test frequency. Best value per column in bold.

All models scored 6.45–7.55 points lower in macro F1 than in weighted F1, showing that rare classes are recognised far less reliably than frequent ones in this long-tailed dataset. The threshold-free metrics ranked the models differently from accuracy: ResNet50 had the highest macro-ROC-AUC (0.9778) and micro-AP (0.7713), ahead of Swin-B (0.9672, 0.7609) and ViT-B/16 (0.9635, 0.7650). ResNet50 therefore ranks the correct class highly even when it is not the top prediction, while the transformers make more correct top-1 decisions.

B. Confidence and Calibration

Correct predictions of all three models had a median softmax confidence close to 0.9 (Fig. 2). This ceiling follows from label smoothing, whose training target for the true class is 0.9 + 0.1/102 ≈ 0.901 (4). Incorrect predictions had a lower median confidence for ResNet50 (about 0.43) than for ViT-B/16 (about 0.50) and Swin-B (about 0.59), and the transformers' incorrect predictions spread up to high confidence: the upper quartile was about 0.80 for ViT-B/16 and 0.85 for Swin-B, against 0.63 for ResNet50. The transformers are thus more likely to be confidently wrong, which makes confidence a weaker signal for rejecting uncertain predictions.

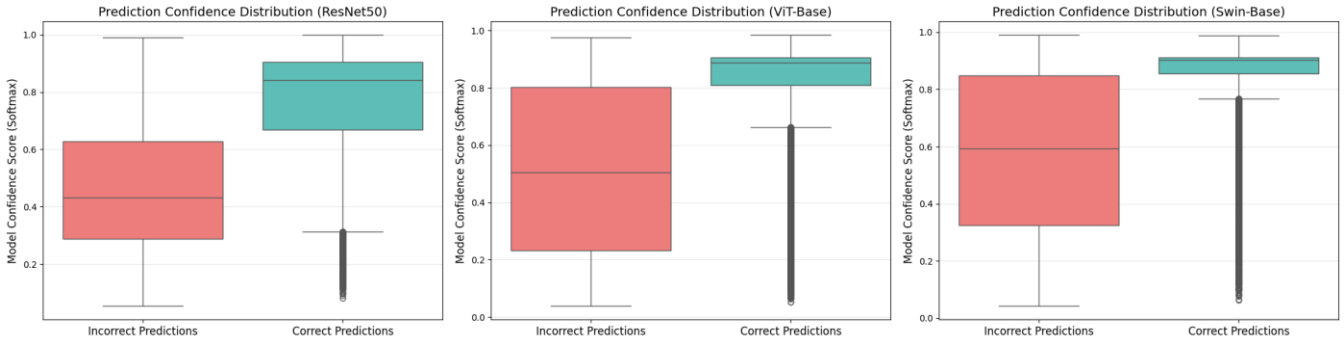


Fig. 2. Softmax confidence of correct and incorrect test predictions for ResNet50 (left), ViT-B/16 (centre) and Swin-B (right).

The reliability diagram confirmed this pattern. ResNet50 followed the diagonal closely across all confidence bins. ViT-B/16 and Swin-B were slightly under-confident below about 0.4 confidence but over-confident in the highest bins: at a mean confidence of about 0.95, the observed accuracy fell to about 0.82 for ViT-B/16 and 0.80 for Swin-B, against about 0.95 for ResNet50. Swin-B was also over-confident between 0.6 and 0.85.

C. Model Agreement

The three models classified 65.2% of test images correctly together, and at least one model was correct on 82.6% (Fig. 3). On 17.4% of images (3,925) all three failed, which marks a set of images that none of the architectures could resolve. A further 7.1% were solved by only one model and 10.3% by two, showing that the architectures make partly different errors.

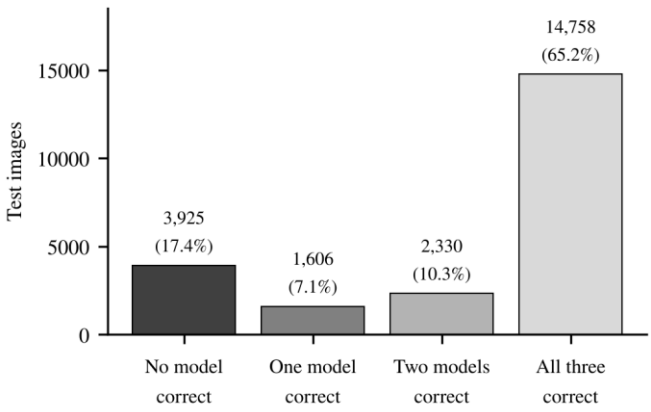


Fig. 3. Number of models (out of three) that classify each test image correctly.

D. Error Analysis

The most frequent confusions were remarkably consistent across architectures. Ten class pairs appear among the fifteen most frequent confusions of all three models (Table V; Swin-B in Fig. 4). Together these ten pairs, out of 10,302 possible ordered pairs, account for 605 (9.3%), 528 (9.6%) and 518 (9.7%) of the errors of ResNet50, ViT-B/16 and Swin-B.

TABLE V
CONFUSION PAIRS SHARED BY ALL THREE MODELS (ERROR COUNTS)

True class → predicted class	ResNet50	ViT-B/16	Swin-B	Relation
tarnished plant bug → Miridae	68	71	67	species → its family
alfalfa plant bug → Miridae	65	51	48	species → its family
Cicadellidae → Cicadella viridis	62	51	62	family → its species
Cicadella viridis → Cicadellidae	50	42	41	species → its family
blister beetle → legume blister beetle	84	55	66	group → member
legume blister beetle → blister beetle	49	63	47	member → group
brown plant hopper → white backed plant hopper	54	54	52	related species
cabbage army worm → beet army worm	50	59	46	related species
large cutworm → black cutworm	66	41	48	related species
rice shell pest → rice leaf roller	57	41	41	similar rice pests
Total (share of all errors)	605 (9.3%)	528 (9.6%)	518 (9.7%)	

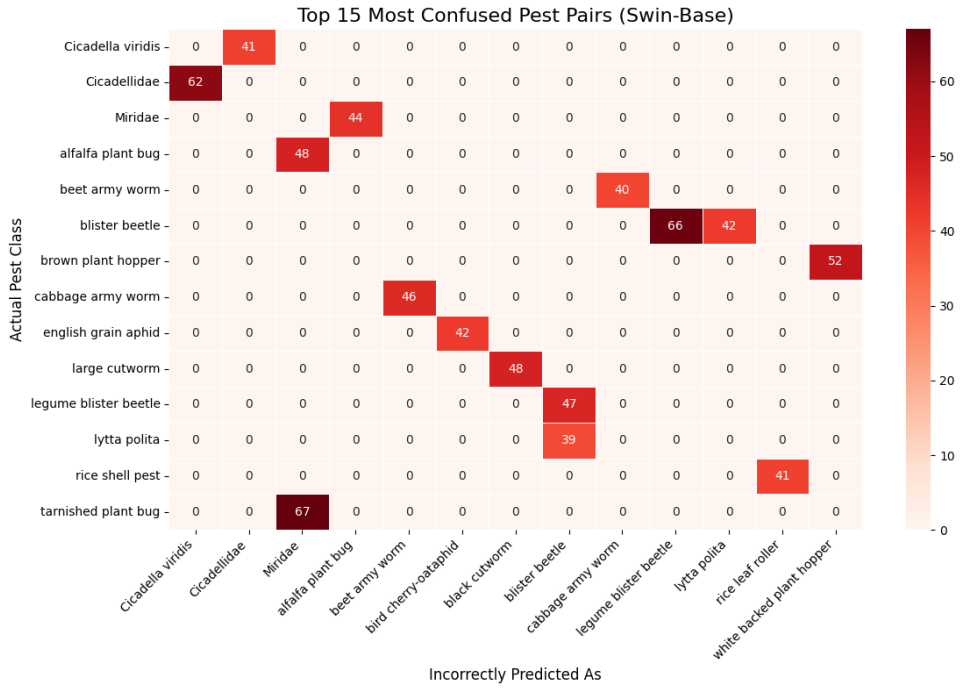


Fig. 4. The fifteen most frequent confusion pairs of Swin-B (true class in rows, predicted class in columns).

Six of the ten pairs are not confusions between distinct taxa of equal rank: one label is a higher taxon and the other belongs to it. Miridae is the plant-bug family that contains both the tarnished plant bug and the alfalfa plant bug, Cicadella viridis is a species of the leafhopper family Cicadellidae, and the legume blister beetle is itself a blister beetle. These six pairs alone account for 378, 333 and 331 errors (5.8–6.2%). An image of a tarnished plant bug labelled as Miridae is arguably correct, so part of the measured error reflects the label taxonomy rather than the models. The remaining pairs, such as brown versus white-backed planthopper and large versus black cutworm, are closely related species that differ in subtle markings.

E. Grad-CAM Visualisation

Grad-CAM heatmaps for the same test image (Fig. 5) show a leaf with rice leaf roller damage overlaid with a website watermark. ResNet50 highlighted the damaged leaf region and ViT-B/16 traced the folded leaf edges, whereas Swin-B placed its strongest activation on the bottom banner of the image. In Fig. 6, ViT-B/16 localised a rice leaf roller larva sharply while Swin-B attended to the image border and leaf edges, and on a field image both ResNet50 and ViT-B/16 focused on the damaged leaf. Across the examples generated,

ResNet50 produced compact, object-centred maps, ViT-B/16 produced the most precise maps, and Swin-B produced the most diffuse maps, sometimes on watermarks, QR codes or borders.

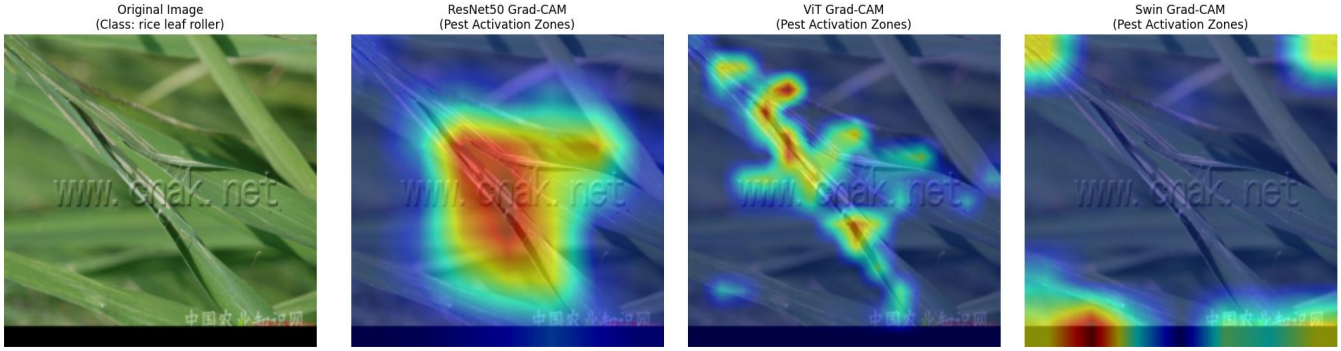


Fig. 5. Grad-CAM for one test image (rice leaf roller) with a website watermark. From left: original, ResNet50, ViT-B/16, Swin-B.

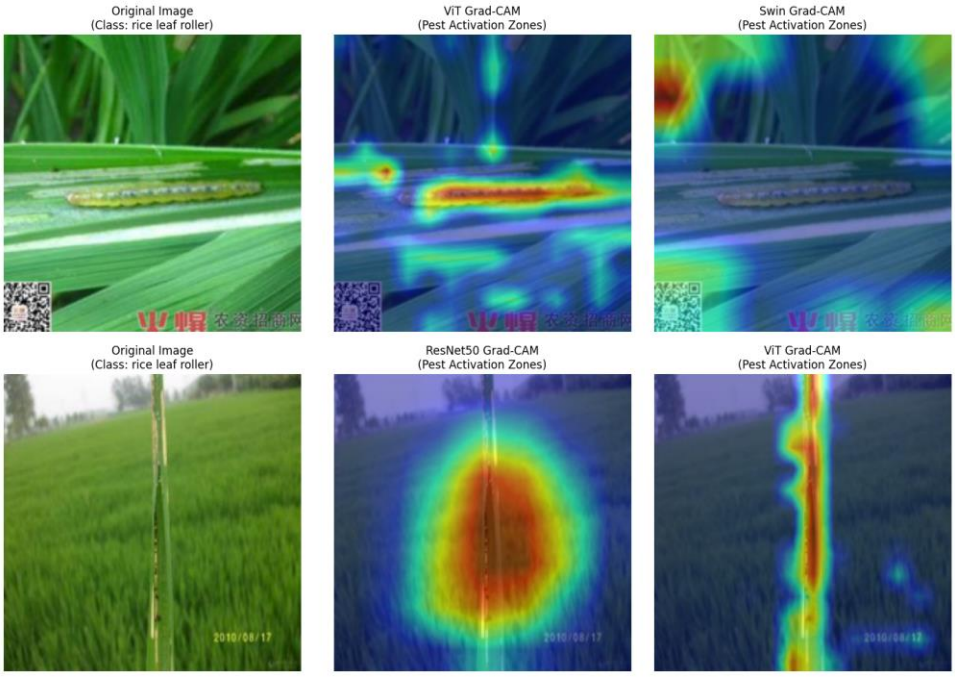


Fig. 6. Further Grad-CAM examples. Top: a rice leaf roller larva (original, ViT-B/16, Swin-B). Bottom: a damaged rice leaf in the field (original, ResNet50, ViT-B/16).

V. DISCUSSION

Transformers are more accurate on fine-grained pest images. Under one identical protocol, both transformers outperformed the CNN by 4.5–5.1 points in Top 1 accuracy and 5.6–6.4 points in macro F1, and the gain was larger for macro than for weighted metrics, which suggests better recognition of less frequent classes. The hierarchical Swin-B was the most accurate model overall, consistent with the view that multi-scale features help when pests occupy small and variable parts of the image [6]. Part of this advantage may also stem from pre-training: the default timm checkpoints for ViT-B/16 and Swin-B were pre-trained on ImageNet-21k/22k before ImageNet-1k fine-tuning, whereas ResNet50 used ImageNet-1k only.

Comparison with published results. Our models fall within the range of recent IP102 results (Table VI). ResNet50 (71.25%) is comparable to the optimised single CNN of [9] (71.84%), and Swin-B (76.38%) exceeds the CNN ensembles of [8] and [10] while remaining below the purpose-built Swin-AARNet [11]. Differences in pre-processing, augmentation and training length between studies mean that these comparisons are indicative.

TABLE VI
COMPARISON WITH PUBLISHED RESULTS ON THE SAME BENCHMARK

Study	Model	Top-1 (%)
Wu et al. [1]	ResNet-50, fine-tuned	49.4
Ayan et al. [8]	GA-weighted CNN ensemble	67.13
Ayan [9]	GA-optimised CNN	71.84
Nanni et al. [10]	CNN ensemble	73.46
Wang et al. [11]	Swin-AARNet	78.77
This work	ResNet50	71.25
This work	ViT-B/16	75.71
This work	Swin-B	76.38

Accuracy and reliability diverge. ResNet50 had the lowest accuracy but the best ROC-AUC, average precision and calibration, while the transformers were over-confident at high confidence. Label smoothing is known to affect calibration and the structure of learned representations [25], and its interaction with each architecture deserves study. For deployment in a pest advisory tool, where uncertain cases should be referred to an expert, calibrated confidence matters as much as accuracy; post-hoc temperature scaling [24] would be a simple remedy for the transformers.

Label quality limits all architectures. The same confusion pairs dominated the errors of all three models, six of the ten shared pairs are parent–child labels in the taxonomy, and 17.4% of test images defeated every model. These observations agree with the label problems documented by Peng *et al.* [12] and indicate that further architectural changes will give diminishing returns unless the labels are cleaned or the class hierarchy is used during training and evaluation. Merging or hierarchically scoring the overlapping labels would give a fairer measure of practical accuracy.

Where the models look. Swin-B was the most accurate model, yet its Grad-CAM maps were the least object-centred and sometimes fell on watermarks and QR codes. Two explanations are possible: Swin-B may exploit dataset artefacts as shortcuts [13], or Grad-CAM on the coarse 7×7 final stage of a windowed transformer may be an unreliable explanation [26]. The heatmaps shown are illustrative examples rather than a quantitative evaluation, and resolving this question requires attribution methods designed for transformers and tests on images with artefacts removed.

Limitations. Each model was trained once with one random seed, so run-to-run variance is not measured. All models used one shared recipe; architecture-specific tuning could change the ranking. The analysis covers one benchmark, and Grad-CAM was examined qualitatively on a small number of images, most from a single class.

VI. CONCLUSION AND FUTURE WORK

This paper compared ResNet50, ViT-B/16 and Swin-B for fine-grained recognition of 102 insect pest classes under an identical training protocol. Swin-B achieved the highest Top-1 accuracy (76.38%), Top-5 accuracy (91.73%) and macro F1-score (69.62%), and both transformers clearly outperformed the CNN in accuracy. ResNet50, however, produced the best-calibrated and best-ranked probabilities. The dominant errors were shared by all architectures and largely arose from overlapping labels in the dataset taxonomy, and 17.4% of test images were misclassified by every model.

Future work will repeat the experiments over several random seeds, apply temperature scaling to calibrate the transformers, train with hierarchy-aware losses that account for the parent–child labels, evaluate on a cleaned version of the dataset, and quantify attribution quality with transformer-specific explanation methods.

DATA AND CODE AVAILABILITY

The benchmark dataset is publicly available from its authors [1]. The Google Colab notebook used for training and evaluation is provided as supplementary material.

REFERENCES

[1] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "IP102: A large-scale benchmark dataset for insect pest recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 8787–8796, doi: 10.1109/CVPR.2019.00899.

[2] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70–90, 2018.

[3] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, Art. no. 2674, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[5] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[6] Z. Liu *et al.*, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012–10022.

[7] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626.

[8] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," *Comput. Electron. Agric.*, vol. 179, Art. no. 105809, 2020.

[9] E. Ayan, "Genetic algorithm-based hyperparameter optimization for convolutional neural networks in the classification of crop pests," *Arab. J. Sci. Eng.*, vol. 49, no. 3, pp. 3079–3093, 2024, doi: 10.1007/s13369-023-07916-4.

[10] L. Nanni, A. Manfè, G. Maguolo, A. Lumini, and S. Brahnam, "High performing ensemble of convolutional neural networks for insect pest image detection," *Ecol. Inform.*, vol. 67, Art. no. 101515, 2022.

[11] X. Wang, Z. Xiao, and Z. Deng, "Swin attention augmented residual network: A fine-grained pest image recognition method," *Front. Plant Sci.*, vol. 16, Art. no. 1619551, 2025, doi: 10.3389/fpls.2025.1619551.

[12] H. Peng *et al.*, "Crop pest image classification based on improved densely connected convolutional network," *Front. Plant Sci.*, vol. 14, Art. no. 1133060, 2023, doi: 10.3389/fpls.2023.1133060.

[13] R. Geirhos *et al.*, "Shortcut learning in deep neural networks," *Nat. Mach. Intell.*, vol. 2, no. 11, pp. 665–673, 2020.

[14] A. Vaswani *et al.*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[15] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021.

[16] R. Wightman, "PyTorch image models," GitHub repository, 2019. [Online]. Available: https://github.com/huggingface/pytorch-image-models

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[18] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 8024–8035.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[21] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[22] P. Micikevicius *et al.*, "Mixed precision training," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[24] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.

[25] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[26] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 782–791.