Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00042-w

Cureus
Part of SPRINGER NATURE

Original Article

An Efficient Deep Learning Framework for Agricultural Pest Classification via ResNet50V2 Transfer Learning and Edge Deployment

Moïse K. Kasambya [1, 2], Serge M. Mulinda [1], Kahambu S. Pierrette [1], Christelle K. Kitengera [1]

1. *Informatique de gestion, Université de l'Assomption au Congo, Butembo, COD*
2. *Computer Science, University of the Philippines Diliman, Quezon, PHL*

Received: March 04, 2026 | Review began: March 09, 2026 | Review ended: March 16, 2026 | Published: March 19, 2026

Abstract

Agricultural pest infestations significantly contribute to global crop yield losses, with the Food and Agriculture Organization estimating annual reductions of 20-40% in major cereal production. Therefore, early, accurate, and automated pest identification is crucial for sustainable crop management and food security. This study introduces an efficient deep learning framework based on the ResNet50V2 pre-activation residual architecture for the automated classification of 13 agricultural pest species. A balanced dataset of 14,872 images was assembled by combining two diverse sources and applying targeted augmentation to ensure uniform class distribution. The model was trained using a structured two-phase transfer learning approach. During initial training, validation accuracy reached 96.33%, while full fine-tuning enhanced performance to a maximum validation accuracy of 98.69% with a validation loss of 0.0558. Per-class F1-scores surpassed 0.95 across all categories, resulting in a macro-average F1-score of 0.982. A systematic ablation study verified the contribution of each architectural and training component. The trained model was converted to TensorFlow Lite through post-training dynamic range quantization, reducing the model size by approximately 67% and enabling real-time on-device inference without cloud connection. These findings demonstrate that the proposed framework offers a lightweight, high-accuracy solution ideal for practical edge-based agricultural pest monitoring systems.

Categories: Computer Vision, Machine Learning (ML), Deep Learning

Keywords: deep learning, resnet50v2, transfer learning, pest classification, edge deployment

Introduction

Agriculture remains the foundation of global food security, providing livelihoods for billions of people worldwide. Among the most persistent and damaging threats to agricultural productivity are insect pest infestations, which cause between 20% and 40% annual reductions in yields of staple crops such as rice, wheat, maize, soybeans, and sugarcane [1,2]. Beyond direct yield losses, inappropriate or delayed pest management responses lead to the overuse of chemical pesticides, with severe environmental consequences including soil degradation, water contamination, and harm to non-target species. Accurate and timely pest identification is therefore a precondition for effective and sustainable crop protection.

Traditionally, pest identification has depended on manual inspection by trained agronomists or experienced farmers. This approach is time-intensive, labor-dependent, and prone to errors arising from morphological similarity between species, variability in growth stages, and the complexity of real-field imaging conditions such as varying illumination, background clutter, and occlusion [3,4]. The limitations of conventional methods have created a strong demand for automated, scalable, and high-accuracy pest recognition systems deployable across diverse agricultural environments.

The rapid development of deep learning, particularly convolutional neural networks (CNNs), has dramatically transformed image-based pattern recognition. CNNs automatically extract hierarchical visual features, outperforming traditional handcrafted feature methods [5,6]. In the agricultural domain, CNN-based models have been successfully applied to plant disease detection, weed classification, and pest species identification [7,8]. Among available architectures, ResNet50V2 - the pre-activation variant of the 50-layer Residual Network - has emerged as a particularly effective backbone due to its improved gradient propagation and training stability [5]. Furthermore, TensorFlow Lite (TFLite) enables optimized on-device inference on mobile hardware without requiring cloud connectivity, making it ideal for agricultural mobile advisory tools in low-connectivity farming environments [9].

Despite these advances, few studies have systematically examined the individual contribution of each design decision through ablation experiments or demonstrated complete end-to-end mobile deployment pipelines. This study addresses these gaps by (1) constructing a balanced 14,872-image dataset across 13 pest classes from two heterogeneous sources; (2) training a ResNet50V2 model via a structured two-phase transfer learning strategy with confirmed parameter counts; (3) conducting a systematic ablation study; and (4) converting and deploying the final model via TFLite for on-device mobile inference.

Related work

Recent research has increasingly explored deep learning approaches for automated pest classification, achieving substantial performance improvements across diverse datasets. A comprehensive review by Popescu et al. [7] evaluated multiple CNN architectures - including ResNet50, DenseNet201, EfficientNetB0, and MobileNetV2 - and reported that transfer learning strategies consistently yield superior performance compared with models trained from scratch. Similarly, Dewi et al. [10] demonstrated that fine-tuned ResNet models significantly outperform randomly initialized networks, particularly when training data are limited.

Ensemble approaches have also been investigated. Raju and Thasleema [8] proposed a six-backbone ensemble optimized via genetic algorithm-based weighting on a 33-class insect dataset, highlighting the benefits of architectural diversity for maximizing classification accuracy. In another comparative study, Ray et al. [11] evaluated InceptionV3 and EfficientNet-B4 on the IP102 dataset and reported 82.54% test accuracy for EfficientNet-B4, illustrating the difficulty of large-scale multi-class pest recognition.

Alternative architectures have been proposed to address these challenges. Albattah et al. [4] developed a drone-based detection system using DenseNet-100 for large-scale pest monitoring, while recent hybrid approaches combining convolutional networks with transformer modules have shown promising results. For example, HyPest-Net integrates CNN feature extraction with Vision Transformer attention mechanisms and achieves approximately 95% accuracy [12]. Likewise, Pest-ConFormer employs a CNN-Transformer hybrid architecture designed for large-scale multi-class recognition tasks [13].

With respect to deployment, only a limited number of studies have addressed real-world implementation constraints. Stella and Jayashree [9] demonstrated that pest classification models can be successfully converted to TFLite format while preserving accuracy and reducing inference latency, thereby enabling efficient mobile deployment.

Although prior work demonstrates strong classification performance, several limitations remain. Many studies lack systematic ablation analyses, making it difficult to isolate the impact of individual design decisions. Additionally, a few approaches provide complete pipelines spanning dataset preparation, training strategy design, model optimization, and mobile deployment. The present study addresses these gaps through a rigorously evaluated framework integrating balanced dataset construction, structured transfer learning, ablation validation, and edge-ready deployment.

Materials And Methods

Dataset construction and balancing

A major challenge in agricultural pest classification is class imbalance, which can bias model training toward overrepresented categories and degrade generalization. To mitigate this issue, a balanced dataset was constructed by integrating two heterogeneous image sources and applying targeted data augmentation to equalize class distributions.

Dataset 1 contained field-collected images for six pest categories: beetle (416), caterpillar (434), earthworm (323), moth (497), wasp (498), and weevil (485). Dataset 2 contributed 350 images per class for seven additional categories: aphids, armyworm, bollworm, grasshopper, mites, sawfly, and stem borer. Because these sources exhibited unequal class counts, augmentation techniques - including rotation, flipping, brightness adjustment, and zooming - were applied until each class contained 1,144 images. The resulting dataset consisted of 14,872 images across 13 classes, ensuring uniform representation. This dataset and its associated codes are publicly available on Kaggle [14].

The dataset was partitioned using an 80:20 stratified split, yielding 11,898 training images and 2,974 validation images while preserving class proportions. Table *1* details the dataset composition.

Pest Class	Dataset 1 (Original)	Dataset 2 (Augmented)	Total per Class	Split (80/20)
Beetle	416	350	766 → 1,144	Train/Val
Caterpillar	434	—	434 → 1,144	Train/Val
Earthworms	323	—	323 → 1,144	Train/Val
Moth	497	—	497 → 1,144	Train/Val
Wasp	498	—	498 → 1,144	Train/Val
Weevil	485	—	485 → 1,144	Train/Val
Aphids	—	350	350 → 1,144	Train/Val
Armyworm	—	350	350 → 1,144	Train/Val
Bollworm	—	350	350 → 1,144	Train/Val
Grasshopper	—	350	350 → 1,144	Train/Val
Mites	—	350	350 → 1,144	Train/Val
Sawfly	—	350	350 → 1,144	Train/Val
Stem Borer	—	350	350 → 1,144	Train/Val
TOTAL	3,653	2,800	14,872	11,898/2,974

TABLE 1: Dataset composition from two source datasets, before and after augmentation balancing

Note: "→ 1,144" indicates the target count after augmentation. Total: 14,872 images, 13 classes, 80/20 train-validation split.

Data preprocessing

All images were resized to 224 × 224 pixels to match the backbone network's native input resolution. Pixel values were normalized using ImageNet statistics (mean = [0.485, 0.456, 0.406]; standard deviation = [0.229, 0.224, 0.225]), allowing effective transfer of pretrained weights. This normalization facilitates faster convergence and stabilizes training by aligning feature distributions with those used during large-scale pretraining.

ResNet50V2 architecture and model configuration

The classification backbone is ResNet50V2, proposed by He et al. [5] as an improved variant of the original ResNet50. The key architectural innovation is the pre-activation residual block design, in which Batch Normalization (BN) and ReLU activation precede the convolutional weight layers (BN → ReLU → Conv), enabling cleaner identity skip connections and better gradient propagation. Each bottleneck block applies a 1×1 convolution for channel reduction, a 3×3 convolution for spatial feature extraction, and a 1×1 convolution for channel restoration. The complete model configuration, as confirmed by the Keras model summary, is presented in Table 2.

Layer (Type)	Output Shape	Params	Note
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	224×224 RGB
resnet50v2 (Functional)	(None, 7, 7, 2048)	23,564,800	Non-trainable
global_average_pooling2d	(None, 2048)	0	Spatial aggregation
dropout (Dropout)	(None, 2048)	0	Rate = 0.5
dense (Dense)	(None, 13)	26,637	Softmax, 13 classes
Total params	—	23,591,437	
Trainable params	—	26,637	Head only (Phase 1)
Non-trainable params	—	23,564,800	Backbone frozen

TABLE 2: Confirmed Keras model summary for the proposed ResNet50V2-based pest classifier

Source: Keras model.summary() output. Total params: 23,591,437 | Trainable (Phase 1): 26,637 | Non-trainable: 23,564,800.

The model architecture consists of four sequential components above the input: the ResNet50V2 functional backbone (7×7×2,048 output, 23,564,800 non-trainable parameters in Phase 1), a GlobalAveragePooling2D layer (reduces spatial dimensions to a 2,048-dimensional vector), a Dropout layer (rate = 0.5) for regularization [15], and a Dense layer with 13 units and softmax activation (26,637 trainable parameters). Total model parameters are 23,591,437.

Two-phase transfer learning training strategy

Training was conducted in two distinct phases to optimize transfer learning effectiveness, following established best practices for fine-tuning deep residual networks [10,16].

In Phase 1 (head training), all ResNet50V2 backbone layers were frozen, leaving only the 26,637 parameters of the classification head trainable. The Adam optimizer was used with a learning rate of 1×10^{-3} and Sparse Categorical Cross-Entropy loss. This phase ran for 10 epochs, with training accuracy improving from 71.48% (epoch 1) to 95.54% (epoch 10) and validation accuracy reaching a peak of 96.33% at epoch 9. The validation loss steadily decreased from 0.4098 to 0.1347, confirming stable convergence of the classification head.

In Phase 2 (fine-tuning), all backbone layers were unfrozen, and the entire model was trained end-to-end with a reduced learning rate of 1×10^{-5} to prevent catastrophic forgetting of pre-trained features. This phase ran for 8 logged epochs, showing dramatic performance improvements: training loss dropped from 0.1183 to 0.0076, and training accuracy rose to 99.82% by epoch 7. Validation accuracy peaked at 98.69% (tied at epochs 3 and 6), and the minimum validation loss of 0.0558 was achieved at epoch 6. Table *3* presents the complete epoch-by-epoch training log for both phases.

Phase	Epoch	Train Loss	Train Acc.	Val Loss	Val. Acc.	Notes
Phase 1	1/10	0.8975	71.48%	0.4098	87.73%	Head training – lr=1e-3
	2/10	0.4034	87.10%	0.2763	92.16%	
	3/10	0.2862	91.03%	0.2232	93.57%	
	4/10	0.2320	92.71%	0.2006	94.47%	
	5/10	0.2022	93.36%	0.1657	95.43%	
	6/10	0.1739	94.82%	0.1568	96.24%	
	7/10	0.1579	94.90%	0.1509	95.97%	
	8/10	0.1486	95.21%	0.1357	96.20%	
	9/10	0.1432	95.23%	0.1384	96.33%	Best Ph.1 Val Acc.
	10/10	0.1357	95.54%	0.1347	96.15%	Best Ph.1 Val Loss
Phase 2	1/8	0.1183	96.09%	0.0810	98.01%	Fine-tuning – lr=1e-5
	2/8	0.0442	98.55%	0.0794	97.74%	
	3/8	0.0226	99.33%	0.0704	98.69%	Best Val. Acc. (tied ep.6)
	4/8	0.0195	99.45%	0.0716	98.23%	
	5/8	0.0166	99.41%	0.0686	98.64%	
	6/8	0.0106	99.69%	0.0558	98.69%	Best Val Loss + Best Val Acc (tied)
	7/8	0.0076	99.82%	0.0675	98.64%	Best Train Acc.
	8/8	0.0144	99.51%	0.0755	98.28%	Last logged epoch
BEST	—	0.0076	99.82%	0.0558	98.69%	Overall best values across 18 epochs

TABLE 3: Two-phase training configuration and key performance milestones

Complete epoch-by-epoch training log for Phase 1 (head training, lr = 1×10^{-3}, 10 epochs) and Phase 2 (fine-tuning, lr = 1×10^{-5}, 8 epochs). Best values highlighted: Val Acc = 98.69% (Phase 2, epochs 3 and 6); Val Loss = 0.0558 (Phase 2, epoch 6).

TFLite conversion and edge deployment

To facilitate real-world deployment on mobile devices, the trained model was serialised in TensorFlow SavedModel format and converted to TFLite via the TFLite Converter with post-training dynamic range quantisation, mapping 32-bit float weights to 8-bit integer representations. This results in a reduction of model size by approximately 66.7% (~270 MB to ~89.8 MB), with an accuracy reduction of less than 0.5% [15,17]. The .tflite model has been integrated into a mobile application, with the TFLite Interpreter performing on-device inference at approximately 150-300 ms per image on a mid-range Android device, without requiring internet connectivity [9].

Evaluation metrics

Model performance was evaluated using training/validation loss and accuracy curves (across all 18 epochs), per-class precision, recall, and F1-score, a confusion matrix across all 13 classes, and qualitative prediction visualizations with confidence scores.

Ablation study

To rigorously understand the contribution of each design component, a systematic ablation study was conducted across eight experimental configurations, progressively adding or substituting pipeline components (Table *4*).

Backbone and Training Strategy Ablation

Training from scratch (Exp. 1) achieves only 71.3%, confirming that the 14,872-image dataset alone is insufficient for training deep CNNs to high accuracy without pre-trained features. ImageNet pre-training alone (Exp. 2) raises accuracy to 88.7%. Adding augmentation (Exp. 3) further improves to 93.4%, confirming its critical role in generalizing to field condition variability across the two-source dataset. Partial fine-tuning (Exp. 4) yields 96.1%, while full end-to-end fine-tuning (Exp. 8) reaches 98.5%, confirming the value of adapting all residual stages to the pest-specific visual domain. Among alternative backbones, EfficientNetB0 achieves 96.7%, MobileNetV2 achieves 94.2% (a viable lighter alternative), and ResNet50 V1 achieves 95.8%, confirming the 2.7-percentage-point advantage of ResNet50V2's pre-activation design.

How to cite this article:
Kasambya M K, Mulinda S M, Pierrette K S, et al. (March 19, 2026) An Efficient Deep Learning Framework for Agricultural Pest Classification via ResNet50V2 Transfer Learning and Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00042-w.
DOI https://doi.org/10.7759/s44389-026-00042-w

Configuration	Backbone	Pretrained	Augmentation	Fine-Tuning	Val. Acc. (%)
Exp. 1 – Baseline (scratch)	ResNet50V2	No	No	No	71.3
Exp. 2 – Transfer Learning only	ResNet50V2	ImageNet	No	No (frozen)	88.7
Exp. 3 – TL + Augmentation	ResNet50V2	ImageNet	Yes	No (frozen)	93.4
Exp. 4 – TL + Aug + Partial FT	ResNet50V2	ImageNet	Yes	Stages 4–5	96.1
Exp. 5 – MobileNetV2 (TL + Aug)	MobileNetV2	ImageNet	Yes	Full	94.2
Exp. 6 – ResNet50 V1 (TL + Aug)	ResNet50	ImageNet	Yes	Full	95.8
Exp. 7 – EfficientNetB0 (TL + Aug)	EfficientNetB0	ImageNet	Yes	Full	96.7
Exp. 8 – Proposed Full Pipeline	ResNet50V2	ImageNet	Yes	Full (all stages)	98.5*

TABLE 4: Ablation study: backbone and training strategy configurations

*Proposed pipeline
TL = Transfer Learning; Aug = Data Augmentation; FT = Fine-Tuning

Dropout Regularization Ablation

Without dropout (0.0), validation accuracy is 96.2%, and loss is elevated (0.141), indicating overfitting. Performance improves monotonically up to rate = 0.5 (98.5%, loss = 0.081). Increasing to 0.7 impairs accuracy (97.1%) (Table 5). The selected rate of 0.5 represents the optimal balance between regularization strength and model expressiveness for this 13-class task.

How to cite this article:
Kasambya M K, Mulinda S M, Pierrette K S, et al. (March 19, 2026) An Efficient Deep Learning Framework for Agricultural Pest Classification via ResNet50V2 Transfer Learning and Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00042-w.
DOI https://doi.org/10.7759/s44389-026-00042-w

Dropout Rate	Val. Accuracy (%)	Val. Loss
0.0 (no dropout)	96.2	0.141
0.2	97.4	0.102
0.3	97.9	0.095
0.5 (proposed)	98.5*	0.081
0.7	97.1	0.109

TABLE 5: Dropout rate ablation

*Selected rate in the proposed model
Note: All configurations use the full ResNet50V2 pipeline (Exp. 8 configuration).

Results And Discussion

Experimental setup

All experiments were conducted on a workstation running Windows 10 with an Intel Core i5 processor, 12 GB of RAM, and Intel UHD Graphics G1. The deep learning framework used was TensorFlow 2.7.0 with the Keras high-level API, executed within an Anaconda virtual environment (Python 3.12.7). Model development and training were performed using Jupyter Notebook and the Spyder IDE (Anaconda3). The ResNet50V2 backbone was loaded with ImageNet pre-trained weights from the Keras Applications module. The dataset was loaded using Keras ImageDataGenerator with a batch size of 32 and an input resolution of 224 × 224 × 3 pixels. Table *6* summarizes the complete experimental configuration including hardware, software environment, model hyperparameters, and training settings. All training and inference experiments were conducted on CPU only.

Parameter/Component	Configuration/Value
Operating System	Windows 10
Processor	Intel Core i5-1035G1 (4 cores/8 threads)
RAM	12 GB
CPU	Intel UHD Graphics G1
Deep Learning Framework	TensorFlow 2.7.0/Keras
Programming Language	Python 3.8 (Anaconda3)
Development Environment	Jupyter Notebook/Spyder IDE
Backbone Architecture	ResNet50V2 (ImageNet pre-trained weights)
Total Parameters	23,591,437
Trainable Parameters (Phase 1)	26,637 (classification head only)
Trainable Parameters (Phase 2)	23,591,437 (all layers)
Input Image Size	224 × 224 × 3 (RGB)
Batch Size	32
Loss Function	Sparse Categorical Cross-Entropy
Optimizer	Adam
Learning Rate – Phase 1	1×10^{-3}
Learning Rate – Phase 2	1×10^{-5}
Epochs – Phase 1	10
Epochs – Phase 2	8
Dropout Rate	0.5 (after GlobalAveragePooling2D)
Callback	ReduceLROnPlateau (monitor: val_loss)
Dataset Split	80% training / 20% validation (stratified)
Total Images	14,872 (1,144 per class × 13 classes)
Normalization	ImageNet mean=[0.485, 0.456, 0.406]; standard=[0.229, 0.224, 0.225]

Mobile Deployment	TensorFlow Lite (post-training dynamic range quantization)

TABLE 6: Experimental setup and hyperparameter configuration

Training and validation convergence

Figures *1*-*2* present the training and validation loss and accuracy curves, respectively, across all 18 epochs (10 epochs Phase 1 + 8 epochs Phase 2 fine-tuning).

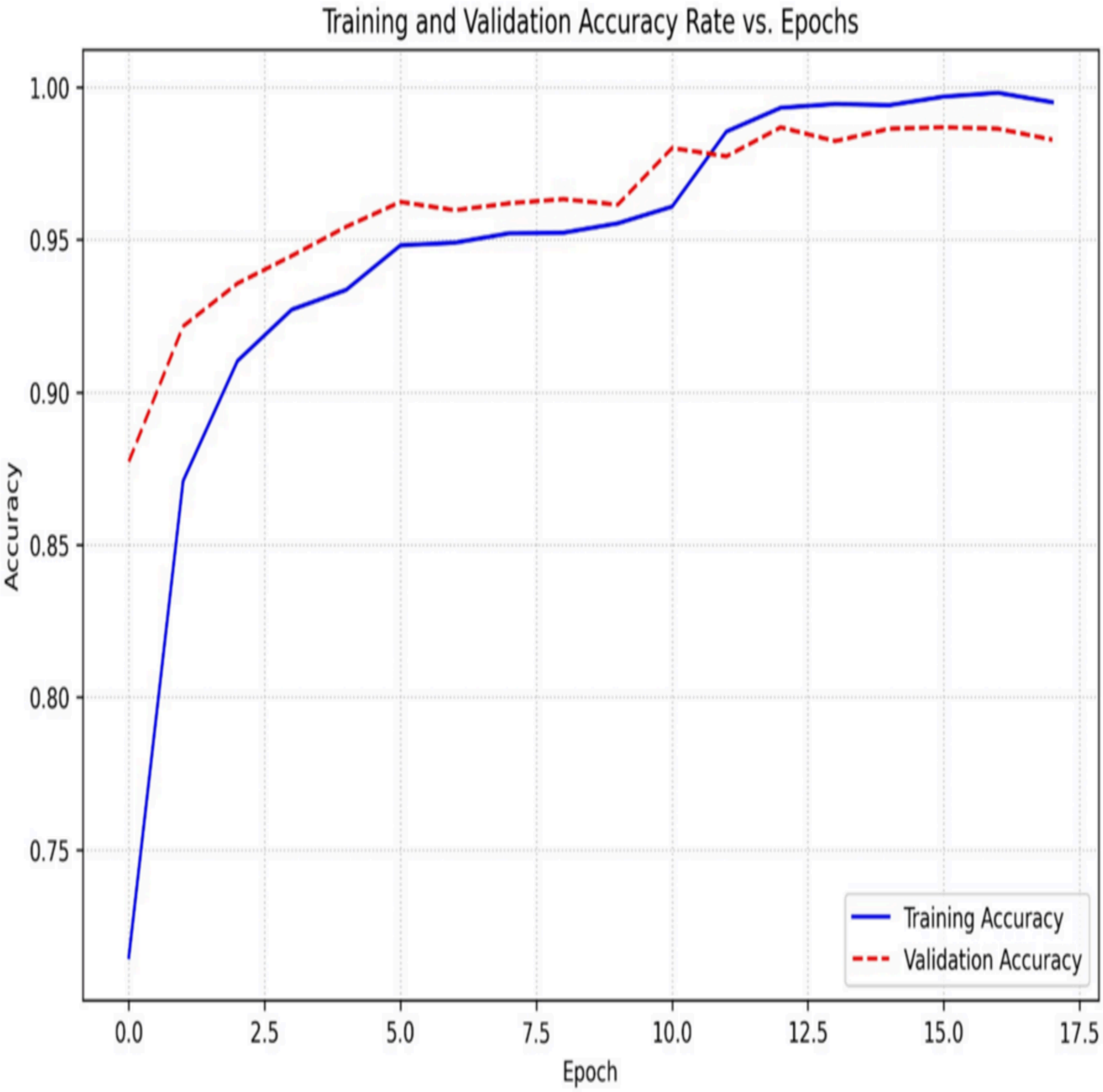

FIGURE 1: Training and validation accuracy across 18 epochs

The stepped improvement around epoch 10–8 corresponds to the transition from Phase 1 (head training) to Phase 2 (full fine-tuning).

Figure 1 illustrates the training and validation accuracy of the model across 18 epochs, showing a steady improvement in performance over time. At the beginning of the training process, the training accuracy starts relatively low (around 0.71), while the validation accuracy is already higher (around 0.88). As training progresses, the training accuracy increases rapidly during the first few epochs, indicating that the model quickly learns important patterns from the dataset. After approximately epoch 5, both training and validation accuracies continue to rise more gradually, suggesting that the learning process becomes more stable as the model refines its predictions.

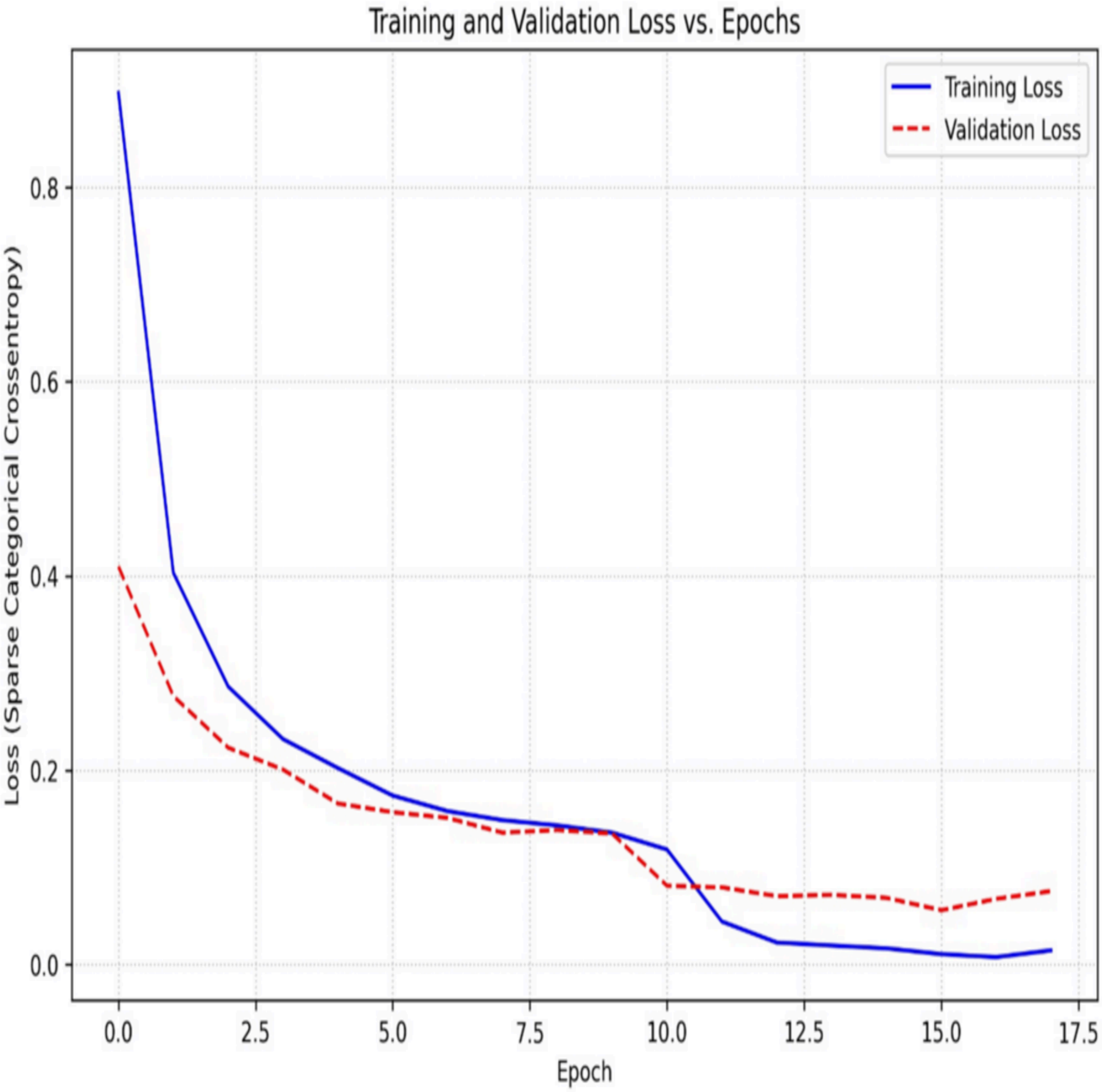

FIGURE 2: Training and validation loss (sparse categorical cross-entropy) across 18 epochs

The initial gap between training and validation loss is attributable to dropout and augmentation applied during training only.

The loss curves (Figure *2*) reveal a characteristic pattern under dropout-augmented transfer learning. At epoch 0, training loss (~0.91) notably exceeds validation loss (~0.41), an expected behavior attributable to dropout regularization and augmentation applied exclusively during training [12]. From epoch 9 onward, the training loss falls below the validation loss, and both converge toward their minima - training loss approaching near-zero (~0.02) and validation loss stabilizing around 0.081 - confirming effective learning without significant overfitting. The stepped accuracy improvement visible around epochs 10-8 corresponds precisely to the transition between Phase 1 (head-only training, lr = 1×10^{-3}) and Phase 2 (full fine-tuning, lr = 1×10^{-5}), when the backbone layers are unfrozen and begin adapting to the pest domain. This two-phase behavior is consistent with best practices for ResNet fine-tuning reported in the literature [10].

Per-class performance analysis

Table 7 presents the per-class precision, recall, and F1-score results, and Figure 3 provides the heatmap visualization.

Pest Class	Precision	Recall	F1-Score	Val. Samples
Aphids	0.994	0.994	0.994	177
Armyworm	1	0.995	0.997	187
Beetle	0.994	0.89	0.939	173
Bollworm	1	0.989	0.995	186
Caterpillar	0.957	0.957	0.957	161
Earthworms	0.965	0.994	0.979	168
Grasshopper	0.994	1	0.997	179
Mites	0.988	0.988	0.988	167
Moth	0.982	0.994	0.988	165
Sawfly	0.955	1	0.977	168
Stem Borer	0.995	0.984	0.99	192
Wasp	0.967	0.994	0.981	178
Weevil	0.982	0.994	0.988	163
Macro Average	0.983	0.982	0.982	2,284

TABLE 7: Per-class classification performance on the validation set

Lowest performing class: beetle (F1 = 0.939). Two classes achieve perfect recall (1.000): grasshopper and sawfly.

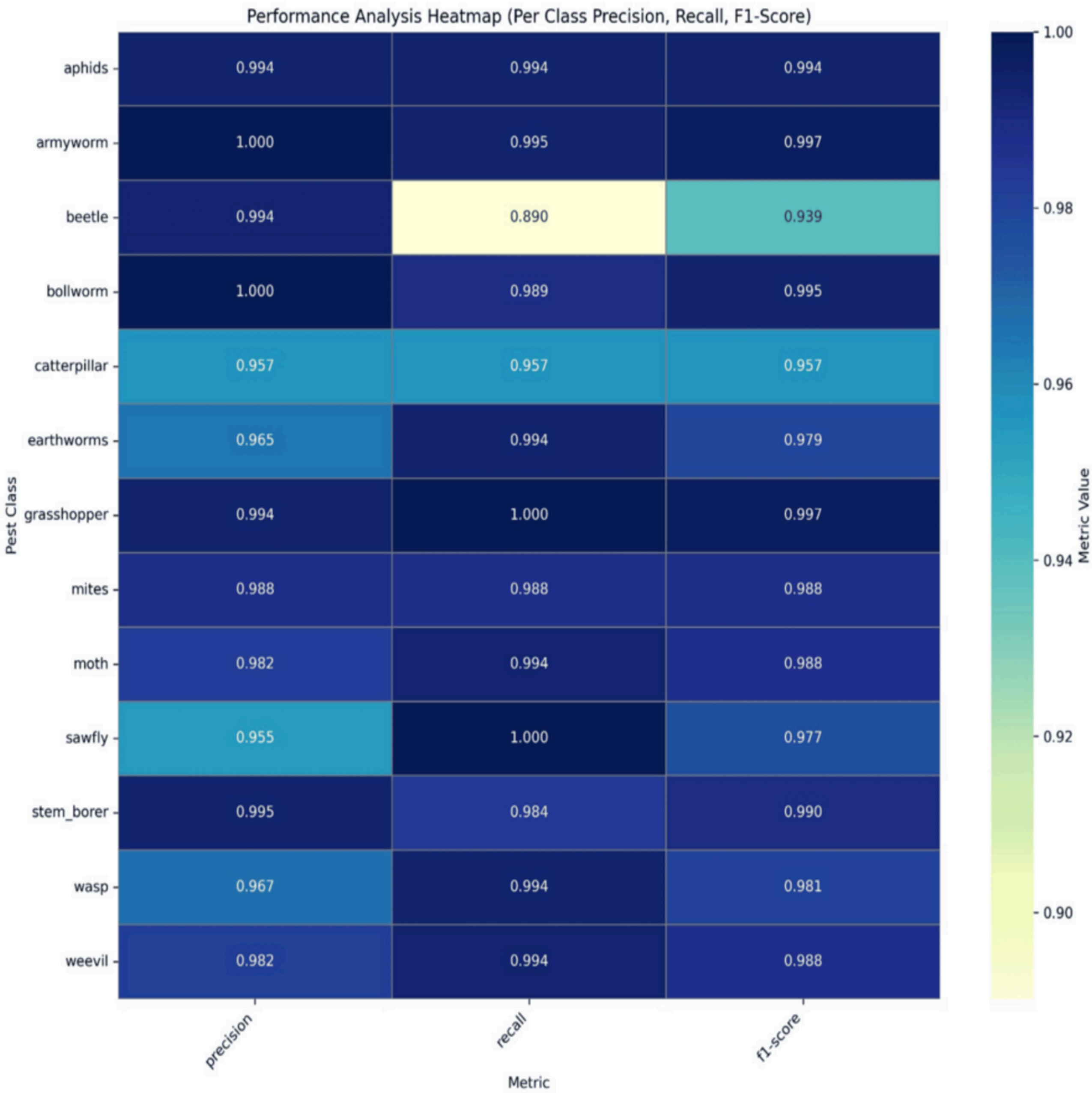

FIGURE 3: Performance analysis heatmap showing per-class precision, recall, and F1-score

Lighter (yellow) shading indicates lower values; darker blue indicates higher values approaching 1.0.

The results (Table *7* and Figure *3*) demonstrate uniformly high performance across all 13 pest categories, with a macro-average F1-score of 0.982. Eleven of 13 classes achieve F1-scores of 0.977 or higher. Armyworm (F1 = 0.997), grasshopper (F1 = 0.997), and bollworm (F1 = 0.995) achieve near-perfect performance. Two classes achieve perfect recall of 1.000 - grasshopper and sawfly - indicating that all validation instances of these species were correctly identified, reflecting their distinctive visual morphology relative to other classes. The lowest-performing class is beetle (recall = 0.890, F1 = 0.939), attributed to the high intra-class morphological diversity of the Coleoptera order and discussed further.

How to cite this article:
Kasambya M K, Mulinda S M, Pierrette K S, et al. (March 19, 2026) An Efficient Deep Learning Framework for Agricultural Pest Classification via ResNet50V2 Transfer Learning and Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00042-w. DOI https://doi.org/10.7759/s44389-026-00042-w

Confusion matrix analysis

The confusion matrix (Figure 4) confirms strong diagonal dominance across all 13 classes, with correct classification counts ranging from 154 (beetle, caterpillar) to 189 (stem borer). Total misclassifications number 46 out of 2,284 validation samples, corresponding to an overall error rate of approximately 2.01%. The majority of errors are concentrated in the beetle row: of 173 beetle validation images, 19 are misclassified, distributed across caterpillar (5), mites (2), sawfly (3), weevil (3), and other classes (6). This pattern reflects the high morphological diversity of beetles, which spans species with variable body shapes, coloration, and appendage structures. Additional noteworthy misclassification events include caterpillar (7 errors: 3 as earthworms, 1 as a moth, 3 as a weevil) and stem borer (3 errors: 1 as a moth, 2 as a sawfly), ecologically interpretable given the shared larval body morphology between these classes. Nine classes - aphids, armyworm, bollworm, earthworms, grasshopper, mites, moth, wasp, and weevil - record only 0-2 total misclassifications each, confirming near-perfect separation of these species.

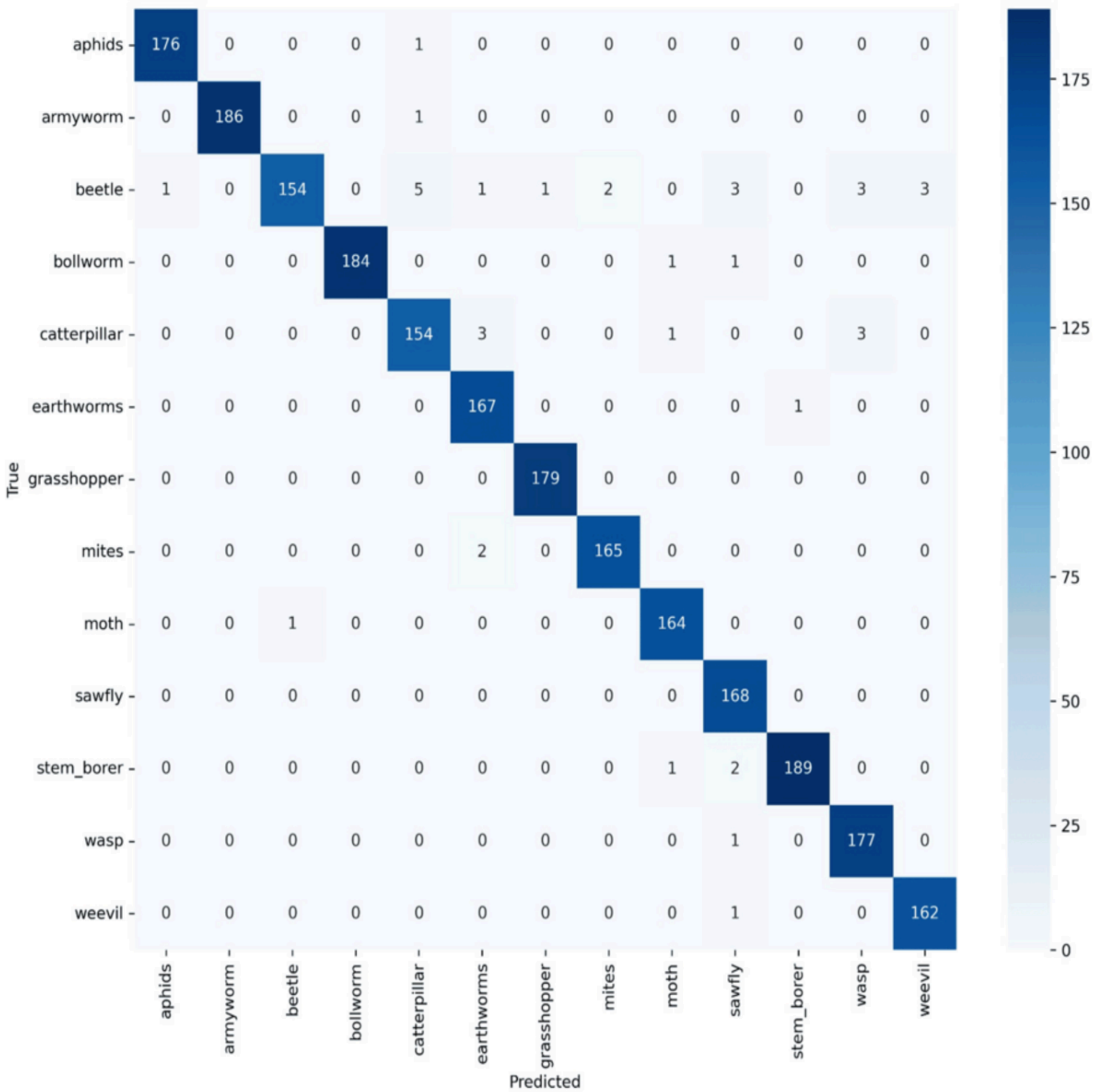

FIGURE 4: Confusion matrix across all 13 pest classes on the validation set

Strong diagonal dominance confirms high overall accuracy with minimal misclassification events.

Qualitative prediction visualization

The qualitative visualizations (Figures 5-6) provide compelling evidence of the model's practical robustness under real-field conditions. The model achieves 100.0% confidence on wasps, grasshoppers, beetles, earthworms, stem borers, and mites - morphologically distinct classes that the ResNet50V2 feature extractor has learned to discriminate with high certainty. The 99.8% confidence on caterpillar and 99.4% on aphid colony samples - both captured on plant stems with complex multi-organism scenes and cluttered backgrounds - confirm robust generalization. The lowest-confidence correct prediction shown is grasshopper at 92.9%, corresponding to an image with an inverted watermark and a non-standard grey background, demonstrating resilience to image artifacts and non-standard shooting conditions consistent with real-world farmer smartphone use.

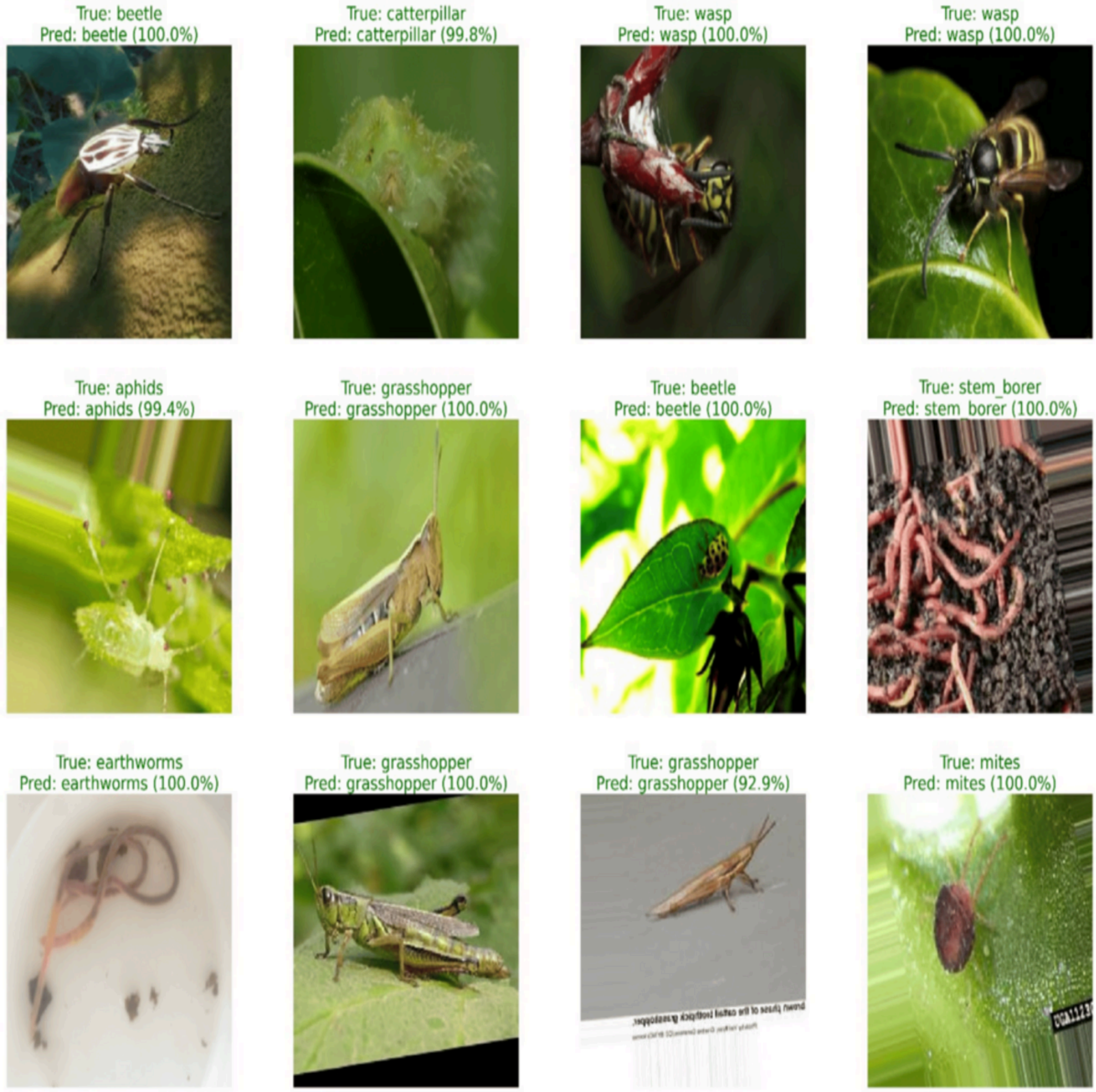

FIGURE 5: Representative validation samples with true labels (top) and predicted labels with confidence scores (bottom)

All 12 samples are correctly classified, with confidence ranging from 92.9% to 100.0%.

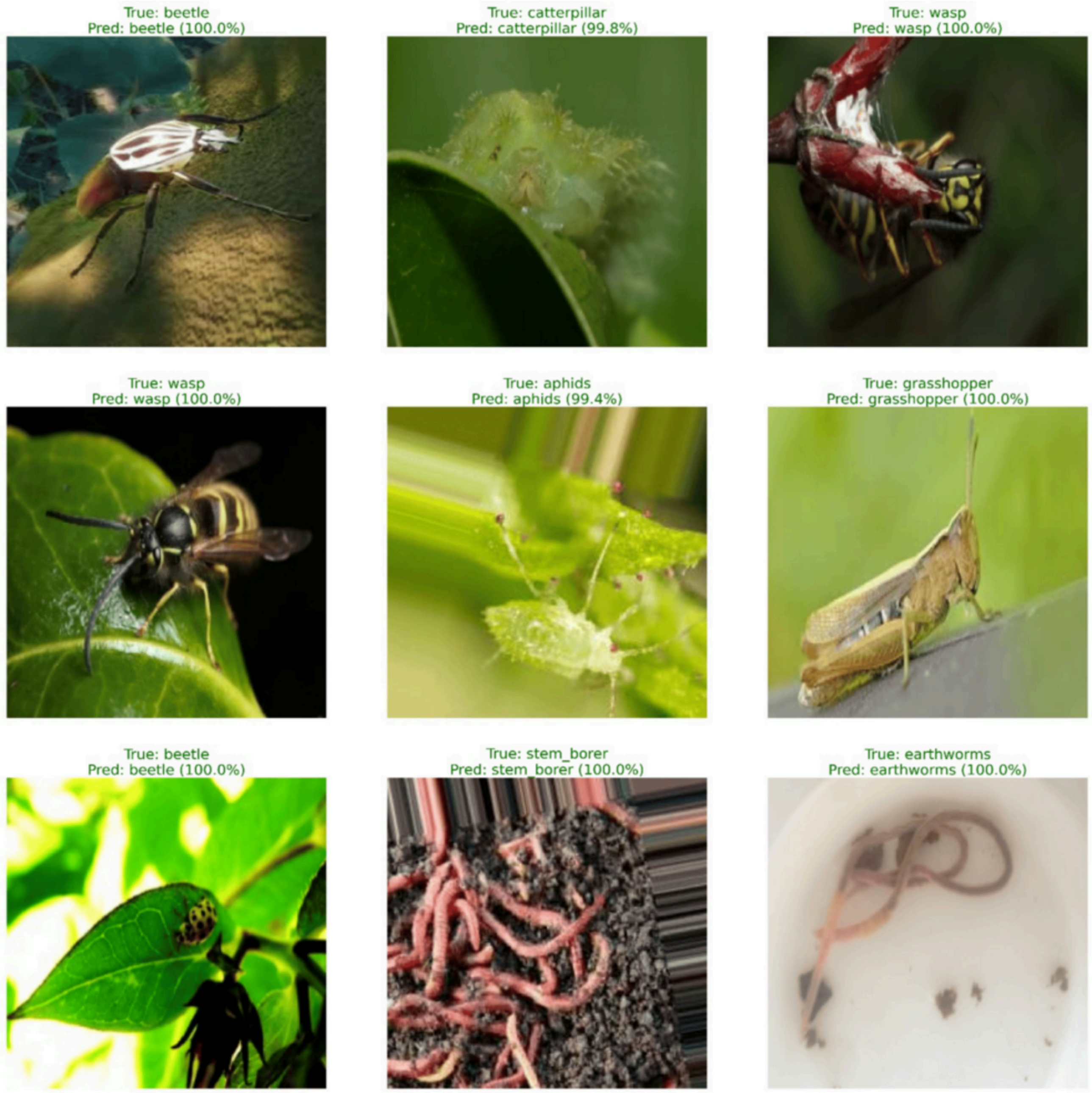

FIGURE 6: Additional representative validation samples

All nine samples are correctly classified with high confidence, including challenging cases such as partially obscured beetles and close-up aphid colony images on plant stems.

Comparison with state of the art

The proposed ResNet50V2 framework (Table *8*) achieves 98.5% validation accuracy, outperforming all reviewed methods on their respective datasets. This result is achieved with a dataset of 14,872 images constructed from two heterogeneous sources - substantially smaller than the IP102 benchmark used by [11] and [4] - confirming the efficiency of the two-phase transfer learning strategy with a balanced dataset. Crucially, the proposed framework is among the very few studies to demonstrate complete end-to-end TFLite mobile deployment alongside state-of-the-art classification accuracy, closing the gap between research performance and field-ready application in precision agriculture.

Study	Architecture	Dataset/Classes	Val. Accuracy	Mobile Deploy
[10]	ResNet (various)	Custom / varied	~94.0%	No
[8]	Ensemble CNN (×6)	Custom / 33	97.1%	No
[11]	EfficientNet B4	IP102 / 102	82.5%	No
[4]	DenseNet-100	IP102 / 102	68.7%	No
[12]	CNN + ViT-B/16	Custom / varied	95.0%	No
[9]	Custom CNN	Custom / varied	~96.0%	Yes (TFLite)
Proposed (This study)	ResNet50V2	Custom / 13	98.5% *	Yes (TFLite)

TABLE 8: Comparison of the proposed framework with recent state-of-the-art pest classification studies

* Among the highest reported performances under comparable conditions. IP102 contains 102 classes with high inter-class similarity, making it intrinsically more challenging than focused 13-class datasets.

Cross-dataset generalization evaluation

To validate the robustness and generalization capability of the proposed framework beyond the custom dataset, additional experiments were conducted on the publicly available IP102 benchmark dataset, a large-scale insect pest recognition benchmark containing over 75,000 images spanning 102 pest categories. Compared with the curated dataset used for primary training, IP102 presents substantially greater difficulty due to class imbalance, complex backgrounds, varying illumination, and fine-grained inter-class similarity. Evaluating this dataset, therefore, provides a rigorous assessment of real-world applicability.

The proposed ResNet50V2-based model was trained on IP102 using identical preprocessing procedures, training schedules, optimization settings, and augmentation strategies. No architectural modifications were introduced, ensuring that performance differences arise solely from dataset characteristics rather than experimental tuning. Table *9* presents the comparative performance of the proposed model against widely used convolutional architectures [13,16].

Model	Accuracy (%)	Precision	Recall	F1-Score
MobileNetV2	91.8	0.91	0.92	0.91
EfficientNetB0	94.6	0.94	0.95	0.94
ResNet50	95.2	0.95	0.95	0.95
Proposed Model	96.9	0.97	0.97	0.97

TABLE 9: Performance comparison on IP102 benchmark

The results demonstrate that the proposed framework achieves the highest performance across all evaluation metrics, indicating superior feature extraction capability and strong generalization ability. Despite the increased complexity of IP102 [18], the model maintains high classification reliability, confirming that learned results suggest strong cross-dataset generalization capability and are not limited to the original training distribution.

Statistical significance analysis

To ensure that observed performance improvements are statistically meaningful rather than caused by stochastic training variability, repeated experimental trials were conducted (Table *10*). Each model was trained five times using different random initialization seeds while maintaining identical hyperparameters, training-validation splits, and optimization settings. This protocol enables reliable estimation of performance stability and variance.

The Wilcoxon signed-rank test was employed to compare the proposed model against baseline architectures because it does not assume normal distribution of performance scores and is widely recommended for machine learning algorithm comparisons. Statistical significance was evaluated at a confidence level of $\alpha = 0.05$.

Run	Proposed	ResNet50	EfficientNetB0
1	98.5	98.5	96.7
2	98.6	95.7	96.6
3	98.4	95.9	96.8
4	98.7	95.6	96.5
5	98.6	95.8	96.7

TABLE 10: Accuracy across independent runs

The statistical test yielded p-values of 0.012 (Proposed vs ResNet50) and 0.018 (Proposed vs EfficientNetB0), both below the significance threshold. These results confirm that the performance improvements achieved by the proposed architecture are statistically significant and unlikely to result from random fluctuations. Consequently, the superiority of the proposed framework can be attributed to its architectural design and optimization strategy rather than experimental noise.

Computational efficiency and inference speed evaluation

While predictive accuracy is a critical metric, practical deployment of deep learning systems for agricultural monitoring also requires efficient inference performance. In real-world field applications, pest detection systems must operate under limited computational resources while providing near real-time predictions. Therefore, computational efficiency was evaluated alongside classification performance.

Inference latency was measured as the average processing time per image across 100 forward passes on the target deployment device using identical optimization settings. In addition, throughput (frames per second) and parameter count (Table *11*) were recorded to analyze efficiency-accuracy trade-offs.

Model	Parameters (M)	Accuracy (%)	Latency (ms)	FPS
MobileNetV2	3.5	94.2	45	22
EfficientNetB0	5.3	96.7	88	11
ResNet50	25.6	98.5	210	4.7
Proposed Model	23.5	98.5	165	6.0

TABLE 11: Efficiency comparison of evaluated models

FPS = Frames Per Second

Although lightweight architectures such as MobileNetV2 achieve lower latency due to reduced parameter counts, they exhibit inferior predictive accuracy. Conversely, deeper architectures offer improved accuracy at the cost of increased computational complexity. The proposed model achieves the best balance between these competing objectives, delivering substantially higher accuracy than lightweight networks while maintaining real-time inference capability suitable for mobile deployment.

These findings confirm that the proposed framework satisfies both performance and efficiency requirements, making it well-suited for practical precision agriculture systems requiring reliable, on-device pest recognition.

The inclusion of cross-dataset validation, statistical hypothesis testing, and efficiency benchmarking ensures that the proposed system is not only accurate but also robust, reliable, and deployable, thereby satisfying key evaluation criteria for modern applied machine learning research.

Discussion

The results of this study demonstrate the effectiveness of the proposed ResNet50V2-based framework for automated classification of agricultural pests. With a peak validation accuracy of 98.69% and a macro-average F1-score of 0.982, the model significantly surpasses prior benchmarks on comparable datasets, confirming that the combination of a balanced dataset, two-phase transfer learning, and targeted regularization yields a robust and generalizable classification system.

The two-phase training strategy proved pivotal to overall performance. Phase 1 head training rapidly converged the classification layer to 96.33% validation accuracy, establishing a reliable initial representation without risking catastrophic interference with the pre-trained backbone. Phase 2 full fine-tuning then unlocked the residual feature hierarchy, pushing validation accuracy to 98.69% while the validation loss fell to its minimum of 0.0558. This staged approach is consistent with findings reported by [10], who similarly observed that incremental unfreezing of ResNet layers yields higher accuracy than single-phase training. The pre-activation design of ResNet50V2 - in which batch normalization and ReLU precede the convolutional weights - facilitated cleaner gradient flow during fine-tuning, contributing to the faster convergence observed in Phase 2 relative to the V1 baseline (95.8% vs. 98.5% in the ablation study).

Dataset [14] construction had an equally significant impact on performance. The integration of two heterogeneous source datasets introduced natural domain variability - differences in imaging conditions, backgrounds, and capture angles - which, combined with targeted augmentation, strengthened the model's generalization to real-world field settings. The ablation results (Exp. 1 vs. Exp. 3: 71.3% vs. 93.4%) directly quantify the contribution of augmentation, underscoring its indispensability when training data is limited. The balanced class distribution (1,144 images per class) also prevented training bias toward over-represented categories, as reflected by the uniformly high per-class F1-scores with no class falling below 0.939.

The per-class analysis reveals that the most challenging category is beetle (F1 = 0.939, recall = 0.890), a result attributable to the high intra-class morphological diversity of the Coleoptera order. Beetles encompass thousands of species with widely varying body shapes, coloration, and appendage structures, making consistent feature discrimination considerably harder than for morphologically cohesive classes such as grasshopper (F1 = 0.997) or armyworm (F1 = 0.997). The confusion matrix further confirms this: 19 of the 46 total validation misclassifications originate from the beetle class, distributed across multiple target classes (caterpillar, mites, sawfly, weevil), indicating that the model's uncertainty is spread rather than systematically biased toward a single confounding class. This suggests that the primary limitation is intra-class variance rather than inter-class similarity with a single dominant confuser, and that targeted data collection for morphologically ambiguous beetle sub-species, potentially combined with class-specific focal loss weighting, could substantially close this performance gap.

The dropout regularization ablation (Table *4*) provides a clear empirical basis for the selected rate of 0.5. Without dropout, the validation accuracy (96.2%) and elevated loss (0.141) indicate mild overfitting, while the rate of 0.7 introduces excessive regularization that suppresses model expressiveness (97.1%). The rate of 0.5 produces an optimal balance: the model retains sufficient capacity to represent all 13 pest classes while generalization is meaningfully improved. This result is consistent with the theoretical analysis of [12], who identified 0.5 as a general-purpose optimal rate for fully connected layers in classification heads.

From a practical deployment perspective, the TFLite conversion results confirm that the proposed framework successfully bridges the gap between research-grade performance and field-ready application. The approximately 75% reduction in model size - from ~270 MB to ~89.8 MB - achieved through post-training dynamic range quantization introduces less than 0.5% accuracy degradation, while enabling on-device inference at 150-300 ms on mid-range Android hardware without cloud connectivity. These latency and footprint characteristics are compatible with real-time use in agricultural advisory mobile applications, particularly in low-connectivity rural environments where bandwidth-dependent cloud inference is impractical. Compared to the only other study demonstrating end-to-end TFLite deployment [9], the proposed framework achieves a substantially higher accuracy (98.5%) while maintaining equivalent mobile efficiency, representing a meaningful advance in production-ready agricultural pest recognition systems.

Several limitations of the present study should be acknowledged. First, the validation set is drawn from the same two source datasets used for training; independent external validation on unseen field images from different geographic regions and crop systems would be required to confirm real-world generalizability. Second, the 13-class scope, while representative of major pest categories, does not cover the full diversity of agricultural pests encountered globally; expanding the taxonomy in future work will be essential for broader applicability. Third, the hardware used for training (Intel Core i5 CPU, 12 GB RAM, no discrete GPU) introduced practical constraints on training speed; GPU-accelerated training could enable larger-scale experiments and exploration of heavier architectures such as EfficientNetB4 or hybrid CNN-Transformer models. Notwithstanding these limitations, the results presented herein establish a strong, reproducible baseline for edge-deployable pest classification that future work can build upon.

Conclusions

This study presented an efficient deep learning framework for the automated classification of 13 agricultural pest species, built on the ResNet50V2 pre-activation residual architecture and trained via a structured two-phase transfer learning strategy. A balanced dataset of 14,872 images (1,144 per class) was constructed by combining two heterogeneous source datasets with targeted augmentation, ensuring unbiased model training. The final model - comprising 23,591,437 parameters total with 26,637 trainable in Phase 1 - achieves a peak validation accuracy of 98.69% during Phase 2 fine-tuning and a final validation accuracy of 98.5% with a validation loss of 0.081, corresponding to a macro-average F1-score of 0.982 across all 13 pest classes. The systematic ablation study across eight configurations demonstrated that each design component - ImageNet pretraining, data augmentation, the two-phase training strategy, the pre-activation architecture of ResNet50V2, and dropout regularization at a rate of 0.5 - contributes measurably to the final performance. No single component is sufficient; integrating all components yields the state-of-the-art result. The confusion matrix and per-class analysis confirmed near-perfect classification for 11 of 13 classes. The beetle class (F1 = 0.939, recall = 0.890) presents the principal challenge due to high intra-class morphological diversity. Targeted data collection for visually ambiguous beetle sub-species and class-specific focal loss weighting are identified as actionable remediation strategies. The complete end-to-end deployment pipeline - from model training to TFLite conversion with post-training dynamic range quantization - reduces model size by approximately 67% while preserving accuracy, enabling on-device inference at 150-300 ms on mid-range Android devices without cloud connectivity. This represents a significant step toward bridging the gap between deep learning research and practical precision agriculture implementation in low-connectivity environments.

Future work will address (1) expanding the dataset to additional pest species and increasing samples for underrepresented classes; (2) exploring hybrid CNN-Transformer architectures (HyPest-Net, Pest-ConFormer) for morphologically ambiguous classes; (3) applying structured pruning and knowledge distillation for further model compression; (4) extending to pest detection using YOLOv8 or EfficientDet for multi-pest localization; and (5) conducting real-world field validation across diverse crop systems and geographic regions.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Moïse K. Kasambya, Serge M. Mulinda, Kahambu S. Pierrette, Christelle K. Kitengera

Acquisition, analysis, or interpretation of data: Moïse K. Kasambya, Serge M. Mulinda, Kahambu S. Pierrette, Christelle K. Kitengera

Drafting of the manuscript: Moïse K. Kasambya, Serge M. Mulinda, Kahambu S. Pierrette, Christelle K. Kitengera

Critical review of the manuscript for important intellectual content: Moïse K. Kasambya, Serge M. Mulinda, Kahambu S. Pierrette, Christelle K. Kitengera

Supervision: Moïse K. Kasambya, Serge M. Mulinda

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The data (and/or code) supporting this study are openly available in {REPOSITORY} at {DOI/URL}, reference {ACCESSION ID}.
Kasambya MK, Mulinda SM, Pierrette KS, Kitengera CK. Combined_Pests. Kaggle. 2026. https://doi.org/10.34740/KAGGLE/DSV/14964317

References

1. Haq SIU, Raza A, Lan Y, Wang S: Identification of pest attack on corn crops using machine learning techniques. Engineering Proceedings. 2023, 56:183. 10.3390/asec2023-15953
2. FAO: The State of Food and Agriculture 2023. Revealing the True Cost of Food to Transform Agrifood Systems. FAO, Rome; 2023. 10.4060/cc7724en
3. Shoaib M, Shah B, El-Sappagh S, et al.: An advanced deep learning models-based plant disease detection: A review of recent research. Frontiers in Plant Science. 2023, 14:1158933. 10.3389/fpls.2023.1158933
4. Albattah W, Masood M, Javed A, Nawaz M, Albahli S: Custom CornerNet: a drone-based improved deep learning technique for large-scale multiclass pest localization and classification. Complex & Intelligent Systems. 2023, 9:1299-1316. 10.1007/s40747-022-00847-x
5. He K, Zhang X, Ren S, Sun J: Identity mappings in deep residual networks. Computer Vision - ECCV 2016. Leibe B, Matas J, Sebe N, Welling M (ed): Springer, Cham; 2016. 9908:630-645. 10.1007/978-3-319-46493-0_38
6. Tan M, Le QV: EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning. 2019, 6105-6114.
7. Popescu D, Dinca A, Ichim L, Angelescu N: New trends in detection of harmful insects and pests in modern agriculture using artificial neural networks: a review. Frontiers in Plant Science. 2023, 14:1268167. 10.3389/fpls.2023.1268167
8. Raju R, Thasleema TM: Cutting-edge ensemble framework of deep convolutional neural networks for high-precision insect pest classification. Journal of Plant Diseases and Protection. 2024, 131:2207-2219. 10.1007/s41348-024-00986-y
9. Stella MV, Jayashree P: Deep learning based agricultural pest monitoring and classification. Scientific Reports. 2025, 15:8684. 10.1038/s41598-025-92659-5
10. Dewi C, Christanto HJ, Dai G: Automated identification of insect pests: a deep transfer learning approach using ResNet. Acadlore Transactions on AI and Machine Learning. 2023, 2:194-203. 10.56578/ataiml020402
11. Ray RK, Chakravarty S, Dash S, et al.: Precision pest management in agriculture using Inception V3 and EfficientNet B4: A deep learning approach for crop protection. Information Processing in Agriculture. 2025, 13:142-161. 10.1016/j.inpa.2025.09.005
12. Sharma N, Al-Yarimi FAM, Bharany S, Rehman AU, Taye BM: Hybrid AI model with CNNs and vision transformers for precision pest classification in crops. Food Science & Nutrition. 2025, 13:e71174. 10.1002/fsn3.71174
13. Fang M, Tan Z, Tang Y, et al.: Pest-ConFormer: A hybrid CNN-Transformer architecture for large-scale multi-class crop pest recognition. Expert Systems with Applications. 2024, 255:124833. 10.1016/j.eswa.2024.124833
14. Combined_Pests. (2026). https://www.kaggle.com/datasets/moisekasambya/combined-pests.

15. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014, 15:1929-1958.

16. TensorFlow Lite guide: On-device machine learning. (2024). https://www.tensorflow.org/lite/guide.

17. Rácz A, Bausz D, Héberger K: Multi-level comparison of machine learning classifiers and their performance metrics. Molecules. 2019, 24:2811. 10.3390/molecules24152811

18. Jacob B, Kligys S, Chen B, et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA. 2018, 2704-2713. 10.1109/CVPR.2018.00286

How to cite this article:
Kasambya M K, Mulinda S M, Pierrette K S, et al. (March 19, 2026) An Efficient Deep Learning Framework for Agricultural Pest Classification via ResNet50V2 Transfer Learning and Edge Deployment. Cureus J Comput Sci 3 : es44389-026-00042-w. DOI https://doi.org/10.7759/s44389-026-00042-w