Deep Learning for Pest Image Classification in Agriculture

Hind Mazin Yassen and Ban Nadeem Dhannoon
Department of Computer Science, College of Science, Al-Nahrain University, 10070 Baghdad, Iraq
hind.m.yassen.sci24@ced.nahrainuniv.edu.iq, ban.n.dhannoon@nahrainuniv.edu.iq.

Keywords: ResNet-50, Pest Detection, IP102, Mixup, SWA, EMA, Label Smoothing.

Abstract: Accurate pest image diagnosis is a critical component of modern precision agriculture. It is important to protect agricultural production, sustainability, and economic progress. Although deep convolutional neural networks, such as ResNet-50, have been widely used for pest classification. However, their performance remains limited when applied to complex, imbalanced, and real-world pest datasets. In this paper, we present an optimized ResNet-50 pipeline for pest image classification on the IP102 dataset, leveraging advanced data preprocessing, targeted regularization strategies, and hyperparameter tuning to improve model robustness and generalization. Moreover, there is substantial bias toward majority classes, and class-imbalance adaptation approaches (e.g., weighted loss functions) have been applied to alleviate this issue. The proposed approach incorporates Mixup data augmentation, label smoothing, exponential moving average (EMA), and stochastic weight averaging (SWA) to enhance training stability and generalization under severe class imbalance. Experimental results demonstrate that the refined ResNet-50 model is a compact and effective CNN that outperforms several related works on the IP102 dataset, achieving 76% accuracy and a weighted F1-score of 0.75. The findings highlight the importance of systematic refinements in enhancing the effectiveness of conventional deep learning models in pest image diagnosis and agricultural tasks.

1 INTRODUCTION

Worldwide, crop damage caused by insects is very high, and significant effort is being invested in developing accurate automated detection systems. It is reported that more than 40% of agricultural harvests are lost each year to pests, pathogens, and weeds, necessitating early and reliable detection for food security [1], [2]. In addition, insect morphology is complex, and many species have a similar appearance, while between instars and habitats, there may be large-scale differences that are difficult to classify using traditional methods [3], [4]. As a result, traditional methods based on machine learning and image processing have been less successful at scale for pest recognition. Agricultural pest identification is an integral part of crop protection; early detection can ensure minimal yield and quality losses. The large-scale pest images dataset IP102 that closely resembles the real-world case with a natural long-tailed distribution, suffers from the challenges of containing intra-class variation, visual similarity among classes, environment noise in real world images, and a severe class imbalance problem [5]. ResNet-50 is deep enough (with 50 layers) to learn rich, hierarchical features at this level of complexity. It is trained and pre-trained on ImageNet and its features (edges, textures, shapes) transfer well for the task of insect recognition. ResNet-50 achieved an accuracy of 49.4% in the IP102 dataset. It remains unclear how effectively relatively simple retrained deep learning models (e.g., ResNet-50) can be improved in pest diagnostics and agricultural applications. Our approach combines customized data enhancement and fine-tuning hyperparameters, enhanced regularization, and adjustments to the loss function to address class imbalance. Despite the considerable progress achieved through architectural innovations such as attention mechanisms and ensemble strategies, limited attention has been devoted to systematically analyzing the impact of optimization techniques on CNN backbones under sever class imbalance conditions. In particular, the IP102 dataset remains challenging due to its high intra-class variability. Therefore, it is still unclear to what extent a carefully optimized conventional architecture can bridge the performance gap without introducing additional architectural complexity. This research aims to address this gab by investigating whether structured optimization strategies can significantly enhance the performance of standard ResNet-50 model on the full 102-class IP102 benchmark. The main contributions of this work, as demonstrated by the experimental results show significantly improvements in accuracy in identifying complex pest data, and are summarized as follows:

1) Label Smoothing helps reduce overconfidence and mitigate overfitting, especially with the IP102 dataset, which exhibits noise and class similarity;
2) The use of the Mixup augmentation technique;
3) EMA (Exponential Moving Average), which maintains a smoothed version of the weights, reducing training fluctuations and often yielding better or more consistent results;
4) SWA (Stochastic Weight Averaging) was enabled after epoch 20 of training, averaging the weights of the last epochs to reach a flatter minimum and improve generalization.

The scientific value of this paper is in methodical combination and empirical confirmation of complementary optimization strategies on the complete 102-class IP102 benchmark. In contrast to previous works where emphasis is mostly laid on architectural changes or attention-based improvements, the present work reveals that structured training optimization in itself can significantly enhance the robustness and generalization of a conventional CNN backbone in case of conditions of severe class imbalance. Additionally, the extensive ablation analysis quantitatively estimates the additive effect of every element of optimization that can offer viable recommendations to applied agricultural AI systems that can be used in real-life settings.

In general, Mixup and Label Smoothing are both regularization techniques. Combining them reduces overfitting and increases robust generalization. EMA and SWA both use weight averaging, but in different ways. This work is organized as follows: related work is detailed in Section 2, the proposed method is presented in Section 3, and the evaluation results are presented in Section 4. Sections 5 and 6 give the discussion and conclusion.

2 RELATED WORKS

Pest identification in real-world agricultural environments poses numerous challenges, including complex backgrounds, varying lighting conditions, visual similarities among categories, and significant discrepancies between categories. Recent surveys and reviews summarize the evolution of features from manual design to deep learning structures and attention-enhancing models, confirming that convolutional neural network (CNN)-based transfer learning remains a solid foundation, while robustness and generalizability under field noise remain challenges [1].

Common standards, such as IP102, provide a large-scale, long-term evaluation environment for insect pest identification and have become a standard testing platform for micro-stage classification methods, achieving an accuracy of 49.4% [2] in 2019. For pest images with the complex background, both residual networks and transferring trails have been used, and some of their extensions to enhance discriminative capability under the condition of noise and background distortion in recent work are adopted, which achieve an accuracy of 55.43% in 2020 [3]. Apart from simplistic underlying architectures, more recent investigations have concentrated on architectural enhancements that aim to better separate out the signals of pesticides from background inference, including adding feature localization or multi- image merging [6], and subsequently they report state-of-the-art performances despite the difficult testing conditions, achieving performance levels comparable to those reported in recent benchmark evaluations: 74.61% accuracy [4] in 2022. Simultaneously, parallel self-attention mechanisms integrated with a convolutional neural network (CNN) and tested on a restricted set of 3,150 images, achieved an accuracy of 99.8% [5] in 2024. Early deep learning-based pest identification systems relied primarily on convolutional structures, with residual learning proving particularly effective due to its optimal stability in deep networks, achieves 67.07% [7] in 2025.

Mixup is a popular data augmentation strategy that linearly interpolates image pairs and their corresponding labels, improving the smoothness of decision boundaries and reducing overshoot, particularly in cases of class imbalance and subtle ambiguity. Mixup improves the accuracy of ResNet-50 Top-1 by approximately 1-2% over ImageNet, with larger gains observed on the ImageNet dataset (2018) [8]. Optimizer design is also an important factor: AdamW separates the weight-decay process from the Adam updates and is typically used to improve generalization compared with simple "L2-as-weight-decay" implementations. AdamW yields around +1% Top-1 accuracy improvement over Adam on ResNet-50 trained on the ImageNet dataset (2019) [9]. Cosine annealing on ResNet-50 with warm restarts consistently improves validation accuracy by approximately 1% compared to monotonic decay schedules on CIFAR 100 / ImageNet datasets (2017) [10]. Weight averaging methods provide an additional boost to generalization: random weight averaging (SWA). SWA increases the accuracy of ResNet-50 by 1–2% by finding flatter minima with better generalization on the ImageNet dataset (2018) [11]. The Exponential Moving Average (EMA) is a weighted averaging method used to stabilize the model. When combined with ResNet-like architectures, EMA consistently yielded improvements of about 0.3–0.6% on ImageNet and between 0.5 and 1.0% on CIFAR-10 in terms of accuracy, while without modifying the architectures found that improved generalization (2018) [12].

Based on these observations, we present an optimized ResNet-50 solver for the IP102 task, emphasizing training stability over architectural sophistication. This methodology uses strong-scaling methods and modern optimization tools to improve generalization and precisely detect pests across broad distributions. This method is very convenient to apply in practice, as it maintains a well-supported infrastructure and reduces the distance from more sophisticated attention-based or convergent models. Although significant progress has been achieved through architectural innovations such as attention mechanisms and multi-scale feature learning, many of these approaches introduce additional computational complexity and require carefully tuned hyperparameters. Moreover, some reported high accuracies were obtained under restricted experimental setting, including reduced class subsets or limited data partitions. In contrast, fewer studies have systematically investigated whether structured optimization strategies alone can substantially improve a standard backbone on the full IP102 benchmark. This gap motivates the present study.

3 MATERIALS AND METHODS

3.1 IP102 Dataset

The Large-scale Insect Pest Recognition IP102 dataset is considered the largest principal benchmark exhibiting poor image quality, and is suitable for real-scene recognition tasks that have 75,222 images (of size 2448x3264 pixels) over 102 different categories [5], [7]. It was used for eight different crops, divided into two main categories: field crops (rice, corn, wheat, beets, and alfalfa) and economic crops (vitis, citrus, and mango).

Figure 1 presents examples of different image types for the insect pest. Its widespread use derives from its realism in simulating real-world complexity, with images partitioned into 45,095 training, 7,508 validation, and 22,619 testing samples [3], [5]. The IP102 dataset presents substantial analytical challenges that standard models fail to overcome.

- Intra-Class and Inter-Class Variation. There are massive visual variations within one species, where the same object can be photographed at various stages from egg to larva, later to pupa, and eventually as an adult, whereas different species can appear almost exactly or identically [3], [5];
- Noise and Imbalance in the Data. Images in the dataset are of varying resolutions, and their backgrounds are complex with a strong class imbalance. Certain classes may be heavily overrepresented (e.g., Chestnut Weevil) or severely underrepresented [7], as shown in Figure 2 shows the distribution of the top 30 most frequent classes in the IP102 dataset. Class indices (1-30) correspond to categories sorted in descending order of sample frequency.

Figure 1: Subclasses of the same insect kind: a) egg, b) larva, c) pupa, d) adult.

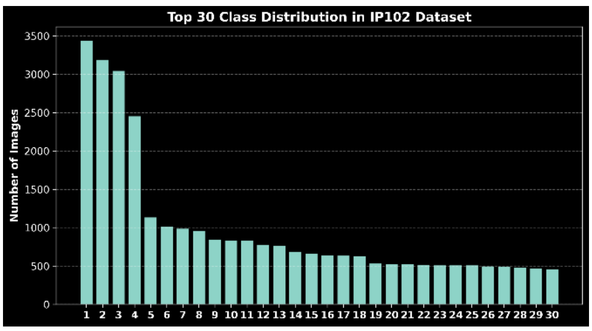

Figure 2: Distribution of the top 30 classes in the IP102 dataset.

3.2 The Proposed Model

The motivation behind our model is to enhance the performance of ResNet-50 on the IP102 pest dataset by embedding both training and architecture optimizations without sacrificing the computational simplicity and efficiency of the base network. Rather than redesigning the model architecture, this work aims at enhancing data processing and regularization through Mixup and Label Smoothing, since their combination reduces overfitting and promotes robust generalization. Using EMA and SWA Weight Averaging also led to increased accuracy on the IP102 dataset, to improve generalization and robustness, and better performance under extreme class imbalance. As shown in Figure 3, the proposed framework enhances the standard ResNet-50 architecture through sequential integration of data augmentation, regularization techniques, adaptive learning rate scheduling and weight averaging strategies.

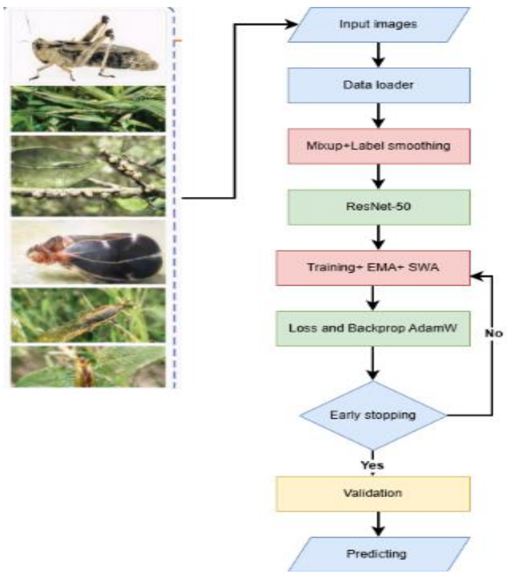

Figure 3: ResNet50 model improvement stages.

The most important matters of new architectural and methodological variations are:

- Keeping the core architecture. The backbone of ResNet-50 was kept making use of its powerful feature extracting properties based on the convolutional and residual modules.
- ImageNet's original classifier in ResNet-50 (1000 classes) was replaced by a new fully connected layer of size 102.

Organization and optimization during training:

- Classifications were smoothed with a cross-entropy by a smoothing coefficient of 0.1 to suppress overconfident predictions and improve generalization on unbalanced data.
- Mixup data augmentation ($\alpha = 0.4$) was utilized in every iteration to promote more continuous decision boundaries and better generalization properties.
- Exponential moving average (EMA) of model weights with a decay rate of 0.999 was applied to stabilize the learning dynamics.
- Stochastic weight averaging (SWA) algorithm was activated after 70% of the training cycles to improve generalization by averaging the weights in the final stage of training.
- The AdamW optimizer with a weight decay value of 0.05 was used, along with the CosineAnnealingLR and ReduceLROnPlateau algorithms, with the addition of a gradient clipping property (max_norm = 5) to stabilize updates.

3.3 Loss Function and Class Imbalance Handling

The IP102 dataset is highly imbalanced, which negatively affects standard cross-entropy training. To address this issue, the proposed model employs:

- Weighted Cross-Entropy Loss, with class weights computed inversely proportional to class frequencies.
- Weighted cross-entropy with classification smoothing was used to address class imbalances and prevent overconfidence.
- This strategy significantly improves performance on underrepresented pest categories.

3.4 Data Preprocessing and Augmentation

To improve generalization and robustness, an extensive data preprocessing and augmentation pipeline is adopted:

- Image resizing to 256 × 256 and random cropping to 224 × 224. To increase data diversity without generating new images, IP102 images are well-suited, as the insect is not always centered, and the background is complex and noisy.
- Random horizontal and vertical flipping.
- Random rotation within ±15 degrees.
- Color jittering (brightness, contrast, saturation).
- Advanced augmentations such as MixUp to reduce overfitting and improve model invariance.

3.4.1 Mix-up Data Augmentation Technique

This technique creates synthetic training samples by linearly interpolating two images and their corresponding labels. In implementation, each input batch is mixed with a randomly shuffled version of itself using a mixing blending coefficient λ sampled from a Beta distribution within the [0, 1] range [13], [14]. The new image is computed as follows:

$$\tilde{x} = \lambda x_i + (1 - \lambda). \quad (1)$$

where x_i, x_j are raw input vectors:

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j. \quad (2)$$

where y_i, y_j *are onehot label encodings.* Where $\tilde{x}$ and $\tilde{y}$ are used as a new training example. This allows the model to learn from a "mixed" version of the data, which can improve generalization by making the model more robust to small variations in input. (x_i, y_i) and (x_j, y_j) are two examples drawn at random from the training data. It is used to:

- Regularizes the model by smoothing decision boundaries.
- reduce overfitting.
- improves generalization on highly imbalanced datasets.
- Forces the model to learn more robust, global features instead of memorizing noise.

3.4.2 Cosine Annealing Learning Rate Scheduler

This schedule adjusts the learning rate during training using a cosine function, as in (3), gradually decreasing it from an initial maximum value to a minimum value over a specified number of epochs or iterations. This allows the model to explore more aggressively at the beginning of training, converge smoothly toward the end, avoid sharp or unstable convergence, and settle into better-performing minima.

$$\eta_t = \eta_{min} + \frac{1}{2} (\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\right)\right). \quad (3)$$

Where:

- ηt is the learning rate at step t;
- ηmax is the initial (maximum) learning rate;
- ηmin is the minimum learning rate (often 0 or very small);
- t is the current step or epoch;
- T is the total number of steps or epochs for annealing.

3.4.3 ReduceLROnPlateau Learning Rate Scheduling Technique

This technique adaptively lowers the learning rate when validation performance stops improving, as in (4), allowing the model to fine-tune more effectively [15], [16]. It is used to prevent training stagnation, to help the optimizer escape regions where the gradient becomes too small, and to allow deeper refinement of learned features.

$$\eta_{\text{new}} = \eta_{\text{old}} \times \text{factor}. \quad (4)$$

3.4.4 Exponential Moving Average (EMA)

EMA is a technique that maintains a smoothed version of the model's weights during training, where this version is updated via an exponential average that gives greater weight to previous weights and reduces the impact of sharp changes resulting from random mini-batch updates, as in (5).

$$w_t(\text{EMA}) = \alpha w_{t-1}^{(\text{EMA})} + (1 - \alpha) w. \quad (5)$$

Where α: 0 < α < 1, and we use α = 0.999.

This strategy reduces the impact of noisy updates, improves training stability, and promotes generalization, particularly on large, unbalanced datasets such as IP102.

3.4.5 Stochastic Weight Averaging (SWA)

SWA improves generalization by averaging the weights from multiple checkpoints collected during late training. Instead of using one final model, SWA computes the mean of several weight vectors [17], as in (6).

$$w_{\text{swA}} = \frac{1}{k} \sum_{i=1}^{k} w_i. \quad (6)$$

It used to:

- Converges toward wider, flatter minima.
- Reduces sensitivity to noise in the optimization landscape.
- Produces smoother decision boundaries and more stable predictions.

To illustrate the qualitative performance of the proposed optimized ResNet-50 model, sample predictions from the test set are presented in Figure 4.

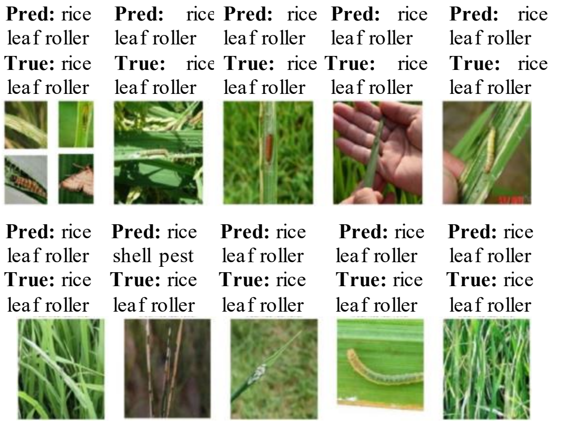

Figure 4: output sample predictions from the optimized ResNet50 Model on IP102 test set. Each image displays the ground truth label and the corresponding predicted label.

3.5 Training Strategy

The model is trained under the following configuration:

- Network Depth. The basic network architecture is based on ResNet-50, which consists of 50 learnable layers, including convolutional layers, residual layers, and a final classification layer;
- Batch size - 64;
- Number of epochs -100. where early stopping (patience = 7) is applied to automatically terminate training when the validation loss no longer improves, preventing overfitting on the IP102 dataset;
- Learning rate - 3×10^{-4};
- Optimizer. AdamW as in (7) [9] with a weight decay coefficient of 0.05.

$$\left.\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 V_{t-1} + (1-\beta_2) g_t^2 \\ \widehat{m}_t &= \frac{m_t}{1-\beta_1^t},\ \hat{v}_t = \frac{v_t}{\alpha-\beta_2^t} \\ \theta_{t+1} &= 1 - \theta_t(Y\eta) - \frac{\widehat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}\eta \end{aligned}\right\} (7)$$

Where:

- θ_t represents the parameters at time step t;
- η is the learning rate;
- $\widehat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates of the first and second moments of the gradients;
- θ_t is a small constant to prevent division by zero;
- λ is the weight-decay coefficient applied directly during parameter updates.

This formulation decouples weight decay from the gradient update step, ensuring consistent regularization without interfering with the adaptive learning rates. The key difference from Adam is that weight decay is applied directly to the parameters rather than being added to the loss function.

- Learning rate scheduler. Cosine Annealing is used to smoothly reduce the learning rate during training;
- Automatic Mixed Precision Training (AMP) to reduce GPU memory consumption without compromising numerical stability.

3.6 Pseudocode for IP102 Insect Classification Pipeline

Algorithm 1: Pseudocode for IP102 Insect Classification Pipeline.

Input: Batch size=64, epochs=100, lr= 3e-4
Output: Classification accuracy
Begin:
Phase 1: Configuration and Data Preparation
Apply data augmentation
Load IP102Dataset and create Train/Val loaders
Phase 2: Model Initialization
Load pre-trained backbone model (ResNet50)
Replace the final classification head (Fully Connected layer) with output size
(number of classes=102)
Phase 3: Training Loop
Initialize: Loss Function (CrossEntropyLoss + Label-Smoothing=0.1),
Optimizer (AdamW)
Initialize Schedulers: CosineAnnealingLR, ReduceLROnPlateau
Initialize: SWA model, EMA model
Apply AMP during forward and backward passes
Clip gradient with max norm=5.0 to stabilize training
Set early stopping variables (patience=7, best_loss)
for epoch = 1 to epochs do
— Training Phase —
for batch (inputs, labels) in training-classes do
Forward Pass: Apply Mixup and compute Mixup loss
Loss Calc: mixed loss = λ×Loss(outputs, labels_a) + (1 −λ)×Loss(outputs, labels_b)
Backward Pass: scale(mixed loss).backward()
Clip gradients (max norm=5.0)
Optimizer step and scaler update
Update EMA model hyperparameters after each optimization step
end for

Table 1: Ablation study using the IP102 dataset

Component Added	Accuracy	Impact on Performance
Baseline (ResNet-50)	49.4%	The initial accuracy of the ResNet50 model [2].
ResNet-50+CE	68.9%	Training of ResNet-50 using CrossEntropy without any enhancements.
Label Modeling	70.2%	Reduces overconfidence and mitigates overfitting.
Mixup	71.6%	Improved generalization by encouraging smoother decisions.
EMA	72.8%	Exponential Moving Average maintains a smoothed version of the weights.
SWA	76%	Stochastic Weight Averaging helped stabilize training and prevent suboptimal convergence.

— Validation Phase —
Evaluate(SWA_model, ValLoader) →val_loss, val_acc
— Scheduler and Checkpoints —
if val loss < best loss then
best loss = val loss
Reset early stopping counter
torch.save(model.state_dict(), config['model_save_path'])
else
Increment early stopping counter
if counter ≥ patience then
Break loop (Early Stopping triggered)
end if
end if
end for
After training: Update_bn(TrainLoader, swa_model)
Return Classification history

3.7 The Ablation Study

To demonstrate the importance of each step in the proposed ResNet-50 architecture on the IP102 dataset, Table 1 shows the accuracy achieved after each step.

3.8 Evaluation Protocol

Accuracy (Acc), macro F1-score, and weighted F1-score were used to evaluate the performance of the proposed model. Accuracy measures the overall proportion of correctly classified samples. Precision reflects the reliability of positive predictions, while recall indicates the model's ability to identify all relevant instances. The F1-score combines precision and recall into a single metric, providing a balanced assessment of classification performance, particularly in datasets with class imbalance. Macro F1 calculates the average F1-score across all classes by treating each class equally, whereas weighted F1 takes into account the number of samples in each class and therefore better reflects performance on imbalanced datasets.

In this study, macro F1 was used to evaluate the model's ability to classify all pest categories equally, while weighted F1 provided an overall assessment considering the actual class distribution. Higher values of these metrics indicate better classification performance and stronger generalization ability of the model.

Previous research on the IP102 dataset has shown varying performance levels for ResNet-50 and its extensions under different experimental conditions. The baseline ResNet-50 achieved 49.4% accuracy and an overall F1-score of 40.1% across 102 categories, highlighting the complexity of the dataset [2]. Subsequent studies improved performance through the integration of deep feature extraction techniques and attention mechanisms [3]. In 2022, ResNet-50 with MS-ALN achieved 74.61% accuracy and an overall F1-score of 67.82% [4]. Although some studies reported accuracy values above 90%, these results were obtained using reduced subsets of classes, limiting direct comparison with full-dataset evaluations [5], [18].

The proposed enhanced ResNet-50 model, incorporating Exponential Moving Average (EMA), Stochastic Weight Averaging (SWA), and adaptive learning rate scheduling, achieved 76.0% accuracy. The obtained macro F1-score of 0.68 demonstrates balanced classification performance across all pest categories, including rare classes, while the weighted F1-score of 0.75 reflects strong overall performance considering the actual class distribution. These results indicate that the proposed approach provides reliable discrimination between pest categories and improves classification robustness on the complete IP102 dataset. All experiments were conducted using the official training, validation, and test split to ensure a fair comparison with previous studies..

4 DISCUSSION

The improved performance of the optimized ResNet-50 model is mainly due to several methodological improvements that specifically tackle the crucial issues of the IP102 dataset, such as enormous intra-class variation, high visual similarity among species, and severe class imbalance.

1) Mixup augmentation helped generalization, since it led to smoother decision boundaries, prevented overfitting to dominant classes, and better enabled handling of rare categories. This can be observed in the rise of macro-level recall and F1-scores.
2) The applied techniques such as Cosine Annealing LR, ReduceLROnPlateau and Stochastic Weight Averaging (SWA) contributed to mitigate training instability and convergence degradation, and produced more uniform predictions over all pest types.

An important fact is that the model is capable of distinguishing very similar looking pests. Overall, the discussion of all those improvements demonstrates that with the help of a combination of methods of sophisticated augmentation, the creation of a more productive optimization schema, and the selection of hyperparameters, ResNet-50 was transformed into a highly competitive pest classification model. A key point to note is that the performance improvements are not due to only one method but the synergetic effect between regularization and weight-averaging methods. The Mixup and Label Smoothing help in providing a smoother decision boundary, with EMA and SWA helping the model to become more stable, as the optimization is directed toward smoother minima. However, the suggested framework is only restricted to image-level classification and does not support localization or real-time detection cases. The future studies can examine how explainability-driven optimization can be integrated to make agricultural decision-support systems even more trustworthy.

5 CONCLUSIONS

In this work, we offer the fine-tuned ResNet-50 architecture for pest recognition on the IP102 dataset. By combining data augmentation methods, improved learning rate scheduling, and stochastic weight averaging, the model achieved significant improvements over previous state-of-the-art results on a ResNet-50 architecture. With an accuracy of 76% and a weighted F1-score of 0.75, our method establishes competitive performance among ResNet-50-based approaches evaluated on the full IP102 dataset. The approach also outperforms the competing ResNet-50 models trained on IP102 without attentional units, multi-image mode, and group learning. The results also make the deployment of lightweight, highly efficient convolutional neural networks for agricultural applications feasible and serve as a practical foundation to real-world pest-diagnosis tools in areas with limited resources. Future work could include incorporating an explainability module, such as Grad-CAM or LIME, extending it to real-time detection systems, and conducting additional assessments of its robustness in field situations.

REFERENCES

[1] B. Guo, J. Wang, M. Guo, M. Chen, Y. Chen, and Y. Miao, “Overview of Pest Detection and Recognition Algorithms,” Electronics, vol. 13, no. 15, p. 3008, Jul. 2024, [Online]. Available: https://doi.org/10.3390/electronics13153008.

[2] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, “IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA: IEEE, Jun. 2019, pp. 8779-8788, [Online]. Available: https://doi.org/10.1109/CVPR.2019.00899.

[3] W. Liu, G. Wu, F. Ren, and X. Kang, “DFF-ResNet: An insect pest recognition model based on residual networks,” Big Data Mining and Analytics, vol. 3, no. 4, pp. 300-310, Dec. 2020, [Online]. Available: https://doi.org/10.26599/BDMA.2020.9020021.

[4] F. Feng, H. Dong, Y. Zhang, Y. Zhang, and B. Li, “MS-ALN: Multiscale Attention Learning Network for Pest Recognition,” IEEE Access, vol. 10, pp. 40888-40898, 2022, [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3167397.

[5] S. M. Hassan and A. K. Maji, “Pest Identification Based on Fusion of Self-Attention With ResNet,” IEEE Access, vol. 12, pp. 6036-6050, 2024, [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3351003.

[6] Y. Chen, M. Chen, M. Guo, J. Wang, and N. Zheng, “Pest recognition based on multi-image feature localization and adaptive filtering fusion,” Frontiers in Plant Science, vol. 14, p. 1282212, Nov. 2023, [Online]. Available: https://doi.org/10.3389/fpls.2023.1282212.

[7] R. Vimala, S. Mehrotra, S. Kumar, P. Kamat, A. Bongale, and K. Kotecha, “GradCAM-PestDetNet: A deep learning-based hybrid model with explainable AI for pest detection and classification,” MethodsX, vol. 15, p. 103533, Dec. 2025, [Online]. Available: https://doi.org/10.1016/j.mex.2025.103533.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in International Conference on Learning Representations (ICLR), 2018.

[9] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in International Conference on Learning Representations (ICLR), 2019.

[10] Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in International Conference on Learning Representations (ICLR), 2017.

[11] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2018.

[12] A. Tarvainen and H. Valpola, "Mean teachers are better role models," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[13] H. Yun et al., "CutMix: Regularization strategy to train strong classifiers with localizable features," in IEEE International Conference on Computer Vision (ICCV), 2019.

[14] C. Si et al., "Better Robustness by More Coverage: Adversarial and Mixup Data Augmentation for Robust Finetuning," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online: Association for Computational Linguistics, 2021, pp. 1569-1576, [Online]. Available: https://doi.org/10.18653/v1/2021.findings-acl.137.

[15] M. Vilar-Andreu, L. García, A. J. Garcia-Sanchez, R. Asorey-Cacheda, and J. Garcia-Haro, "Enhancing Precision Agriculture Pest Control: A Generalized Deep Learning Approach With YOLOv8-Based Insect Detection," IEEE Access, vol. 12, pp. 84420-84434, 2024, [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3413979.

[16] A. Khedher, A. Ramadhanty, and M. I. Fanani, "State-of-the-Art CNN Models for Plant Disease Classification Using Transfer Learning," J. SUS (Jurnal Sains dan Teknologi), vol. 33, no. 6, pp. 1637-1650, Oct. 2025.

[17] H. Guo, J. Jin, and B. Liu, "Stochastic Weight Averaging Revisited," Applied Sciences, vol. 13, no. 5, p. 2935, Feb. 2023, [Online]. Available: https://doi.org/10.3390/app13052935.

[18] B. Ramalingam et al., "Remote Insects Trap Monitoring System Using Deep Learning Framework and IoT," Sensors, vol. 20, no. 18, p. 5280, Sep. 2020, [Online]. Available: https://doi.org/10.3390/s20185280.

[19] Z. Anwar and S. Masood, "Exploring Deep Ensemble Model for Insect and Pest Detection from Images," Procedia Computer Science, vol. 218, pp. 2328-2337, 2023, [Online]. Available: https://doi.org/10.1016/j.procs.2023.01.208.