Classification of agricultural pests through digital images using deep learning

Omitted due to the double-blind review

***Abstract*—In the agricultural sector, pest detection is vital, and given the susceptibility of human analysis to errors, deep learning solutions such as Convolutional Neural Networks (CNNs) provide a promising alternative. Classifying insect pests is challenging due to the high variability among species across different regions and their various life stages. In this study, we evaluate several deep learning models and training strategies for automatic pest image classification. We analyze four CNN architectures—AlexNet, ResNet-50, EfficientNet, and Vision Transformer (ViT). Following a hyperparameter optimization step, the models were fine-tuned, and we examined the impact of four data augmentation strategies on classification performance using the Agricultural Pests dataset. ViT demonstrated superior performance, achieving an accuracy of 0.9574 without data augmentation. Although ViT did not benefit from data augmentation, these techniques proved essential for enhancing the performance of AlexNet, ResNet-50, and EfficientNet. Our findings underscore the potential of deep learning methods for pest classification, offering valuable tools to help professionals maintain crop quality and value.**

***Index Terms*—Pest classification, deep learning, data augmentation, hyperparameter optimization**

I. Introduction

Agriculture is fundamental to economic development and human nutrition, providing a stable and reliable food supply essential for global food security. Beyond food production, agriculture stimulates economic growth and creates jobs, especially in rural areas [1]. Each year, crop yields are affected by various natural disasters, and one of the most significant factors influencing crops is the impact of insect pests. Detecting, identifying, and providing timely feedback at the early stages of pest infestation is crucial to addressing this issue [2].

Recently, agriculture has transformed with machine learning (ML) and deep learning techniques (DL). These technologies provide valuable insights into crop health, moisture content, and soil quality. As a result, resources can be used more efficiently, reducing waste and increasing yields. Additionally, the use of image recognition algorithms aids in the early detection of pests, diseases, and nutrient deficiencies in crops [1].

Computer-aided diagnosis (CAD) systems that employ Deep Learning models, such as Convolutional Neural Network (CNN)s, enable the extracting of rich hierarchical features from images, establishing classification models that can match or even surpass the accuracy of human visual analysis, analyzing and classifying hundreds of samples within a few minutes [3] [4]. However, training these models is complex, as it requires a large number of images, careful adjustment of hyperparameters, data augmentation procedures, and the application of transfer learning. Therefore, experiments are essential to develop suitable models and training strategies for each problem [3].

Classifying insect pests presents a significant challenge in visual recognition due to the wide variability among insect species that affect different countries, regions, and crops. Additionally, many insects undergo distinct transformations throughout their life cycle, such as eggs, larvae, pupae, and adult stages, complicating their identification. In some cases, the insects remain concealed within crops, making them detectable only through the damage they cause to plants. These complexities underscore the necessity of developing advanced transfer learning techniques to create models with robust generalization capabilities [5].

For these reasons, this paper proposes a method for automatically classifying biotic agents to improve the efficiency of pest classification using Deep Learning techniques. This work aims to analyze the performance of CNN models in classifying images of pests in crop lines. We evaluated the AlexNet, ResNet-50, ViT, and EfficientNet architectures and compared the impact of four data augmentation strategies on the prediction performance. All this after performing a grid-search-based hyperparameter optimization.

This paper is organized as follows. In Section II, we present the related works. In Section III, we described the material and methods used, and in Section IV, the results are presented and discussed. Finally, we conclude the paper in Section V.

II. Related Work

Ullah et al. [6] propose the DeepPestNet framework for classifying ten classes of crop pests from Deng's crop dataset. They employed data augmentation techniques, including image rotations, and verified the generalization capability of their model with the Kaggle Pest Dataset.

Khanramaki et al. [7] propose the recognition of three common citrus pests, including citrus Leafminer, Sooty Mold, and Pulvinaria. They proposed an ensemble of CNN classifiers

considering three levels of diversity (classifier, feature, and data), as well as data augmentation techniques to improve model generalization.

Mota et al. [5] conducted a comparative analysis of three deep learning models—ResNet, AlexNet, and Vision Transformer (ViT)—for classifying the IP102 dataset. The study explored the effects of standard data augmentation combined with CutMix across different configurations of the dataset, focusing on the 10% and 20% most frequent classes within the field crop and economic crop groups, as well as combined groups. The authors found that transformer-based architectures outperformed CNN-based models when trained with CutMix. Specifically, ViT models improved their performance, while AlexNet's performance declined, and ResNet-50 showed stable results when CutMix was applied.

Wang et al. [8] present a new approach called InsectMamba for insect classification that integrates state-space models, convolutional neural networks, multi-head self-attention, and multi-layer perceptrons. The proposal enables comprehensive feature extraction through the main characteristics of the respective models. The authors evaluated InsectMamba on five different datasets: Farm Insects, Agricultural Pests, Insect Recognition, Forestry Pest Identification, and IP102, which allows quantifying the generalization ability of the approach in terms of accuracy and F1 score. The results outperformed other approaches and proved the importance of each component of the proposed model.

Uttarwar et al. [9] propose a new deep learning-based model for classifying leaf diseases and agricultural pests based on the EfficientNetB0 architecture. The evaluation in accuracy terms reaches 100%, which demonstrates the ability of this type of technique to perform the task. Huang [10] compares fine-tuned models for pest classification. The authors evaluated ResNet50, Xception, EfficientNet B0, and MobileNe models regarding accuracy, F1 score, recall, and precision on the Agricultural Pests Image Dataset. In summary, the EfficientNet model achieved an accuracy of 88.26% on the task with a low computational cost.

Bhatt et al. [11] present an approach incorporating features from four different CNNs (ResNet50, Xception, DenseNet121, and EfficientNetB5) for improved feature extraction in the agricultural pest classification task. The authors also used the Agricultural Pests Image Dataset to evaluate the proposal, and in terms of accuracy, they obtained 89.35% on the test set. Finally, Abdulrazzaq et al. [12] performed experiments with deep learning-based models on the Agricultural Pests Image Dataset using data augmentation. The approach's results were generally satisfactory in accuracy, precision, recall, and F1 score, emphasizing accuracy, with values close to 94% in the test set.

III. Material and methods

A. Dataset

The Agricultural Pests Dataset was selected to evaluate our training strategies. It contains 5,494 images, organized into 12 classes: Ants, Bees, Beetles, Caterpillars, Earthworms, Earwigs, Grasshoppers, Moths, Slugs, Snails, Wasps, and Weevils. The dataset is available on the Kaggle platform[1], with images sourced from Flickr and resized to a maximum dimension of 300 pixels. It offers diverse shapes, colors, and sizes, making it ideal for developing and evaluating machine-learning models for pest classification challenges. We used the dataset version already split into training and test sets. Figure 1 shows one sample from each class, and Table I provides further details.

TABLE I
Information about the Agricultural Pests Dataset.

Class	Total	Train	Test
Ants	499	400	99
Bees	500	405	95
Beetles	416	331	85
Caterpillars	434	329	105
Earthworms	323	246	77
Earwigs	466	390	76
Grasshoppers	485	390	95
Moths	497	397	100
Slugs	391	316	75
Snails	500	405	95
Wasps	498	392	106
Weevils	485	394	91

B. Architectures

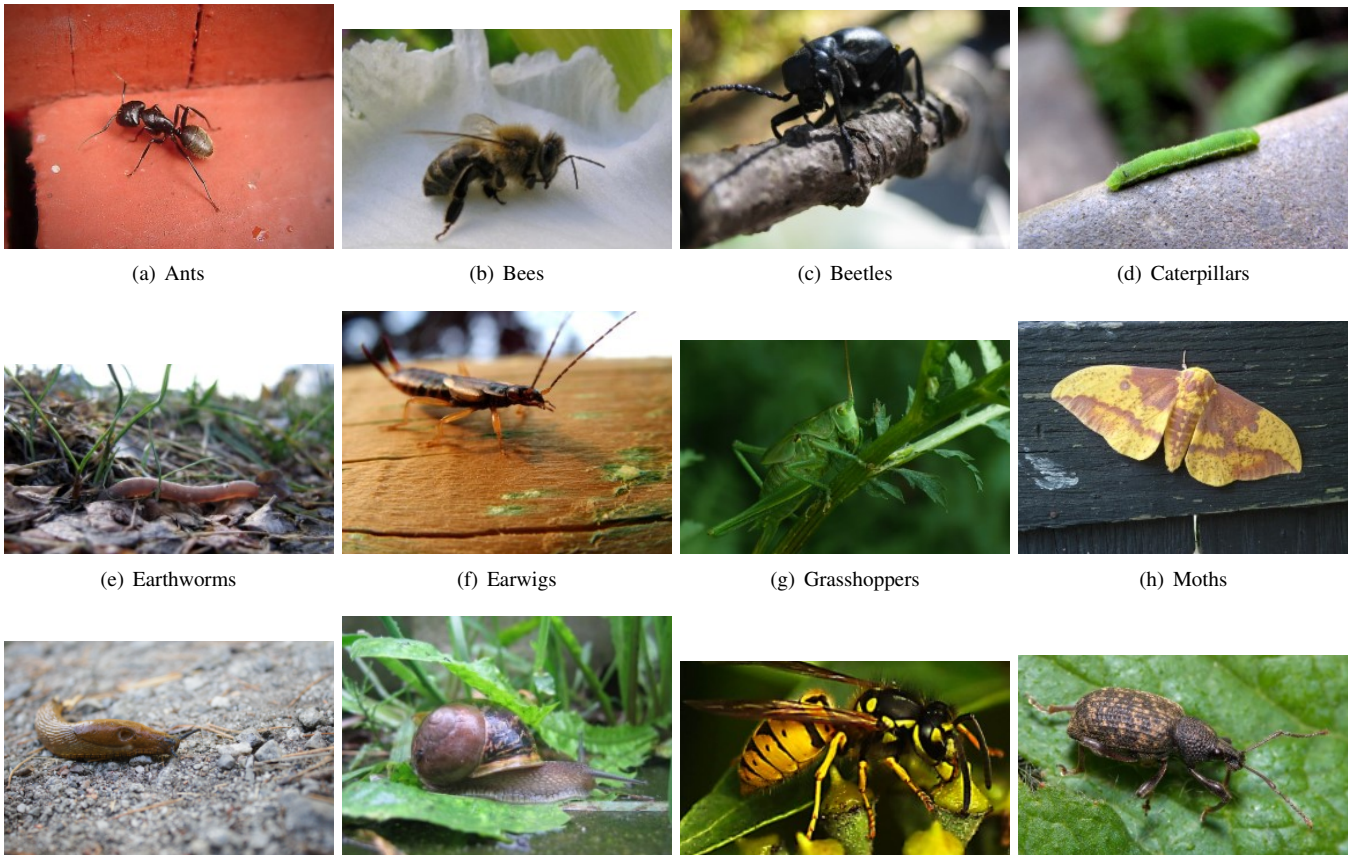
The architectures used in this work were AlexNet, ResNet-50, EfficientNet, and Vision Transformer (ViT), an attention-based architecture. All models were fine-tuned using the models obtained from torchvision, which was pre-trained using ImageNet [13].

AlexNet [14] is a pioneering architecture that significantly contributed to advancements in deep learning. It won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. This architecture used multiple convolutional layers (five convolutional layers followed by three fully connected layers), the Rectified Linear Unit (ReLU) activation function, which significantly sped up the training process, and other features that made it stand out as a classic architecture for deep learning.

ResNet-50 [15] is a 50-layer architecture that revolutionized deep learning through its innovative use of residual learning. The architecture addresses the vanishing gradient problem, enabling the training of very deep networks by incorporating shortcut connections or skip connections within residual blocks.

EfficientNet [16] is a family of architectures designed to achieve state-of-the-art performance while being highly efficient in terms of computational resources. The key innovation

[1]https://www.kaggle.com/datasets/gauravduttakiit/agricultural-pests-dataset



(a) Ants (b) Bees (c) Beetles (d) Caterpillars

(e) Earthworms (f) Earwigs (g) Grasshoppers (h) Moths

(i) Slugs (j) Snails (k) Wasps (l) Weevils

Fig. 1. One sample from each class of the Agricultural Pests Dataset dataset.

behind EfficientNet is using a compound scaling method, which uniformly scales the network's depth, width, and resolution. This balanced scaling results in better performance and efficiency than traditional scaling methods.

Vision Transformer (ViT) [17] is an innovative architecture for image recognition that adapts the Transformer model, originally developed for natural language processing (NLP), to computer vision tasks. ViT leverages self-attention mechanisms to capture global relationships within an image, providing a compelling alternative to traditional convolutional neural networks CNN.

C. Data augmentation

Data augmentation is a technique usually applied to small datasets. It involves artificially increasing their size and diversity during training by applying various transformations to the original images. This method helps improve the generalization ability of models by providing them with more varied examples, thus reducing overfitting and making the models more robust to variations in the input data.

In this work, we explored several data augmentation strategies to improve the model's performance.

The strategy involved no data augmentation, called No DA, in which the images were processed with only a random resized crop operation to a 224 $\times$ 224 pixels size. This baseline was used to assess the impact of augmentation techniques on the model's performance.

The first data augmentation strategy, DA-1, includes a random horizontal flip and a random rotation of up to 15 degrees. Images were also subjected to random resized crop operations to a final size of 224 $\times$ 224 pixels, with a scaling factor between 0.8 and 1.0. Furthermore, a color jitter transformation modifies the brightness, contrast, saturation, and hue of the images. The brightness, contrast, and saturation values were varied by a factor of 0.2, while the hue was adjusted by a factor of 0.1. Finally, a random erasing operation was applied with a probability of 50%, where small portions of the image (from 2% to 25%) were randomly erased.

In the second strategy, DA-2, we slightly modified the previous augmentation pipeline. While the random horizontal flip, rotation, and resized crop remained the same, the hue adjustment was removed from the color jitter transformation. The random erasing operation remained unchanged.

Finally, the third strategy (DA-3) resembles the DA-2 strategy. However, the random erasing operation was excluded, and the remaining transformations were applied as in the previous strategy.

These different data augmentation strategies allowed us to

assess the impact of each transformation on the model's overall performance, intending to achieve the best trade-off between accuracy and generalization.

D. Experiment design

First, we randomly split 20% of the Agricultural Pest Datast dataset's training set to build a validation set in a stratified way.

Considering the validation set, we optimize the batch size (BS) and learning rate (LR) hyperparameters using a grid-search strategy with the following search spaces:

- LR : {0.01, 0.001, 0.0001, 0.00001}
- BS : {16, 32, 64, 128}

Then, we trained each architecture described in Section III-B using the Adam optimizer, considering the best values for batch size and initial learning rate obtained through the hyperparameter optimization. The training consists of fine-tuning the models available in the torchvision library. Each architecture was fine-tuned without data augmentation and with each data augmentation strategy described in Section III-C.

During training, we reduce the learning rate when the validation loss stabilizes, i.e., it didn't improve for a certain number of epochs, 10 for this work. We also early stopped the training when the validation loss did not improve over 21 consecutive epochs.

E. Computational environment

The experiments were executed on a PC with a 3.00 GHz Core I5 CPU and 32 GB of RAM equipped with a GPU NVIDIA GTX 1080 Ti. The PC was running Linux Ubuntu 20.04 LTS, and the experiments were developed using Python 3.9, PyTorch 2.0.1, torchvision 0.15.2 with CUDA Toolkit 10.1, Scikit-learn 1.2.2, and Matplotlib 3.7.1.

F. Model evaluation

We used four traditional classification metrics to evaluate the experimental setup: accuracy, precision, and F1 score. These metrics enable understanding the model's capability to extrapolate the knowledge learned to unseen samples. These metrics are defined in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Equations 1-4 summarize each metric considered:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F1 - Score = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{4}$$

IV. Results and discussion

Table II shows the results of the hyperparameter optimization performed through grid search over the search space described in Section III-D. The table presents the set of hyperparameter values that resulted in the best validation accuracy for each model.

TABLE II
Optimized hyperparameter values for each architecture.

Architecture	BS	LR	Acc. Val.	Epochs
AlexNet	32	0.00001	0.8168	19
ResNet-50	32	0.0001	0.9820	14
EfficientNet	16	0.0001	0.9930	9
ViT b 16	64	0.0001	0.9910	28

Table III presents each model's accuracy, considering each architecture and data augmentation strategy evaluated on both the validation and test sets. The best results across different models are highlighted in bold, while results across various data augmentation strategies are italicized. In terms of test accuracy, it can be observed that the ViT model outperformed the others, regardless of the data augmentation strategy. However, it did not benefit from data augmentation, as its highest accuracy was achieved without it, with 0.9572. EfficientNet gets its best test accuracy when trained with data augmentation strategy DA-1, while REsNe-50 and Alexnet with DA-2 strategy.

Tables V and IV summarize each model's precision, recall, and F1 score, evaluated on both the validation and test sets using macro and weighted averages, respectively. The macro average is the arithmetic mean of each class's metrics, while the weighted average accounts for the class proportions when computing the overall metric. The results in these tables align with those in Table III. Vision Transformer (ViT) models outperformed all others, achieving the best results without any data augmentation strategy. Specifically, ViT's precision, recall, and F1-scores, reported in pairs for weighted and macro averages (weighted, macro), were (0.9574, 0.9562), (0.9572, 0.9548), and (0.9571, 0.9553), respectively.

Regarding precision, recall, and F1 scores, ResNet-50 and EfficientNet-B4 performed best when trained with DA-2 and DA-1 data augmentation strategies. AlexNet, despite being a classic architecture, showed the best performance with DA-1 for these metrics, though it performed better in accuracy with DA-2. Overall, AlexNet delivered the weakest results among the models, which was anticipated given its older architecture. Its inclusion in the study highlights the complexity of the classification task.

In Figure 2, we may visualize the validation and test accuracy for each architecture when trained with all data augmentation strategies.

V. Conclusion

In this study, we compared several deep learning architectures for classifying pest images and assessed their performance with different data augmentation strategies. The

TABLE III
EXPERIMENT RESULTS WITH THE MODELS TRAINED WITH DIFFERENT DATA AUGMENTATION STRATEGIES (ACCURACY).

		VAL. Acc.				TEST Acc.			
	Architecture	**No DA.**	**DA-1**	**DA-2**	**DA-3**	**No DA.**	**DA-1**	**DA-2**	**DA-3**
Acc.	**AlexNet**	0.8168	0.8191	0.8134	0.8180	0.8044	0.8098	*0.8144*	0.8107
	ResNet-50	0.9352	0.9374	0.9397	0.9363	0.9308	0.9263	*0.9381*	0.9227
	EfficientNet b4	0.9499	0.9534	0.9556	0.9488	0.9245	*0.9436*	0.9363	0.9190
	ViT b 16	0.9613	0.9590	0.9590	0.9511	***0.9572***	**0.9454**	**0.9500**	**0.9472**

TABLE IV
EXPERIMENT RESULTS WITH THE MODELS TRAINED WITH DIFFERENT DATA AUGMENTATION STRATEGIES (WEIGHTED AVERAGE).

		VAL.				TEST			
	Architecture	**No DA.**	**DA-1**	**DA-2**	**DA-3**	**No DA.**	**DA-1**	**DA-2**	**DA-3**
Prec.	**AlexNet**	0.8169	0.8198	0.8093	0.8155	0.8094	*0.8135*	0.8116	0.8110
	ResNet-50	0.9350	0.9377	0.9395	0.9367	0.9320	0.9270	*0.9383*	0.9226
	EfficientNet b4	0.9500	0.9539	0.9564	0.9505	0.9244	*0.9437*	0.9362	0.9202
	ViT b 16	0.9615	0.9590	0.9595	0.9510	***0.9574***	**0.9456**	**0.9503**	**0.9474**
Rec.	**AlexNet**	0.8168	0.8191	0.8134	0.8180	0.8044	0.8098	*0.8144*	0.8107
	ResNet-50	0.9352	0.9374	0.9397	0.9363	0.9308	0.9263	*0.9381*	0.9227
	EfficientNet b4	0.9499	0.9534	0.9556	0.9488	0.9245	*0.9436*	0.9363	0.9190
	ViT b 16	0.9613	0.9590	0.9590	0.9511	***0.9572***	**0.9454**	**0.9500**	**0.9472**
F1	**AlexNet**	0.8147	0.8139	0.8181	0.8137	0.8039	0.8052	*0.8103*	0.8054
	ResNet-50	0.9346	0.9363	0.9388	0.9358	0.9301	0.9255	*0.9375*	0.9216
	EfficientNet b4	0.9494	0.9522	0.9553	0.9478	0.9239	*0.9429*	0.9356	0.9179
	ViT b 16	0.9612	0.9586	0.9588	0.9506	***0.9571***	**0.9452**	**0.9496**	**0.9471**

TABLE V
EXPERIMENT RESULTS WITH THE MODELS TRAINED WITH DIFFERENT DATA AUGMENTATION STRATEGIES(MACRO AVERAGES).

		VAL.				TEST			
	Architecture	**No DA.**	**DA-1**	**DA-2**	**DA-3**	**No DA.**	**DA-1**	**DA-2**	**DA-3**
Prec.	**AlexNet**	0.8044	0.8128	0.7986	0.8052	0.7984	*0.8063*	0.8049	0.8030
	ResNet-50	0.9314	0.9332	0.9356	0.9318	0.9290	0.9240	*0.9370*	0.9193
	EfficientNet b4	0.9473	0.9519	0.9536	0.9471	0.9215	*0.9424*	0.9331	0.9166
	ViT b 16	0.9584	0.9558	0.9557	0.9483	***0.9562***	**0.9434**	**0.9471**	**0.9464**
Rec.	**AlexNet**	0.8067	0.8057	0.7996	0.8028	0.7974	0.8017	*0.8044*	0.8035
	ResNet-50	0.9301	0.9334	0.9343	0.9309	0.9278	0.9240	*0.9364*	0.9209
	EfficientNet b4	0.9450	0.9504	0.9527	0.9450	0.9224	*0.9405*	0.9346	0.9173
	ViT b 16	0.9578	0.9555	0.9561	0.9481	***0.9548***	**0.9441**	**0.9478**	**0.9453**
F1	**AlexNet**	0.8032	0.8031	0.7956	0.8007	0.7946	0.7971	*0.8018*	0.7974
	ResNet-50	0.9302	0.9319	0.9341	0.9306	0.9269	0.9227	*0.9359*	0.9190
	EfficientNet b4	0.9455	0.9496	0.9523	0.9439	0.9214	0.9407	*0.9332*	0.9153
	ViT b 16	0.9578	0.9552	0.9554	0.9478	***0.9553***	**0.9434**	**0.9468**	**0.9456**

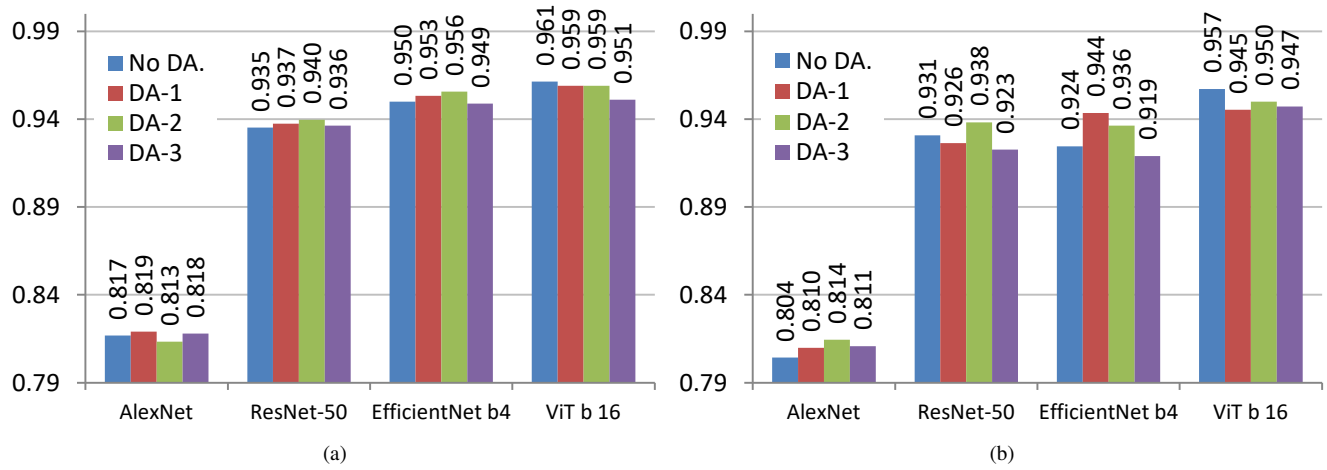


Fig. 2. Charts with the accuracies obtained over the validation (a) and test (b) sets through the performed experiments.

models were fine-tuned using pre-trained models, and hyper-parameter optimization was carried out for each architecture. Our results indicate that machine learning and deep learning techniques are effective for pest image classification, with the Vision Transformer (ViT) achieving the highest test accuracy of 0.9572 when trained without any data augmentation. ViT outperformed all other architectures across all evaluated metrics (accuracy, precision, recall, and F1 score). Although ViT did not benefit from data augmentation strategies, these techniques were crucial for the strong performance of other architectures. ResNet-50 and EfficientNet-B4 achieved their best results when trained with DA-2 and DA-3, respectively. These findings highlight the feasibility of applying deep learning methods for pest classification and lay the groundwork for developing accessible, efficient tools to aid professionals in identifying crop pests.

Future research will involve testing additional deep learning architectures and exploring new training strategies, such as enhanced data augmentation techniques. As insect pests are diverse worldwide, and some pests present very different appearances in each life stage, we intend to use other datasets and exploit advanced transfer learning techniques.

ACKNOWLEDGMENT

Omitted due to the double-blind review

REFERENCES

[1] S. Palei, R. K. Lenka, S. S. Nayak, R. Mohanty, B. Jena, and S. Saxena, "Precision agriculture: Ml and dl-based detection and classification of agricultural pests," in *2023 2nd International Conference on Ambient Intelligence in Health Care (ICAIHC)*. IEEE, 2023, pp. 1–6.

[2] W. Zhang, X. Xia, G. Zhou, J. Du, T. Chen, Z. Zhang, and X. Ma, "Research on the identification and detection of field pests in the complex background based on the rotation detection algorithm," *Frontiers in Plant Science*, vol. 13, p. 1011499, 2022.

[3] R. G. P. Neto, P. M. de Sousa, L. F. R. Moreira, P. I. V. G. God, and J. F. Mari, "Enhancing green coffee quality assessment through deep learning," in *Anais do XVIII Workshop de Visão Computacional*. SBC, 2023, pp. 84–89.

[4] J. G. Esgario, P. B. de Castro, L. M. Tassis, and R. A. Krohling, "An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 38–47, 2022.

[5] G. S. de Lima Mota, L. H. Silva, L. F. R. Moreira, and J. F. Mari, "Classifying pests in crop images using deep learning," in *Anais do XVIII Workshop de Visão Computacional*. SBC, 2023, pp. 42–47.

[6] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, "An efficient approach for crops pests recognition and classification based on novel deeppestnet deep learning model," *IEEE Access*, vol. 10, pp. 73 019–73 032, 2022.

[7] M. Khanramaki, E. A. Asli-Ardeh, and E. Kozegar, "Citrus pests classification using an ensemble of deep learning models," *Computers and Electronics in Agriculture*, vol. 186, p. 106192, 2021.

[8] Q. Wang, C. Wang, Z. Lai, and Y. Zhou, "Insectmamba: Insect pest classification with state space model," *arXiv preprint arXiv:2404.03611*, 2024.

[9] M. Uttarwar, G. Chetty, M. Yamin, and M. White, "An novel deep learning model for detection of agricultural pests and plant leaf diseases," in *2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. IEEE, 2023, pp. 1–6.

[10] Q. Huang, "Comparison of deep transfer learning models for pest image classification in agriculture," *Highlights in Science, Engineering and Technology*, vol. 94, pp. 9–16, 2024.

[11] A. Bhatt, D. Patel, V. Ajmeri, and P. Goel, "Deep transfer learning based light-weight agricultural pest classification model using convolutional neural networks," in *2024 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2024, pp. 175–180.

[12] A. R. Abdulrazzaq, A. K. TüRkben, and S. Kurnaz, "Detection and classification of agricultural pest and vermin's using full-convolutional neural network," in *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*. IEEE, 2023, pp. 1–6.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.