

Classification of Agricultural Pests Through Digital Images Using Deep Learning

Classificação de Pragas Agrícolas por meio de Imagens Digitais Utilizando Deep Learning

Pedro Lucas de Oliveira Costa¹, Thiago Matheus de Oliveira Costa¹,
Larissa Ferreira Rodrigues Moreira¹, Leandro Henrique Furtado Pinto Silva¹,
João Fernando Mari^{1*}

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.

Keywords: Pest Classification — Deep Learning — Data Augmentation — Hyperparameter Optimization

Resumo: No setor agrícola, a detecção de pragas é vital, e, dada a suscetibilidade da análise humana a erros, soluções baseadas em deep learning, como Redes Neurais Convolucionais (CNNs), oferecem uma alternativa promissora. A classificação de pragas de insetos é desafiadora devido à alta variabilidade entre espécies em diferentes regiões e aos seus diversos estágios de vida. Neste estudo, avaliamos vários modelos de deep learning e estratégias de treinamento para a classificação automática de imagens de pragas. Analisamos quatro arquiteturas de CNNs—AlexNet, ResNet-50, EfficientNet e Vision Transformer (ViT). Após uma etapa de otimização de hiperparâmetros, os modelos foram finamente ajustados, e examinamos o impacto de quatro estratégias de aumento de dados no desempenho da classificação, utilizando o Agricultural Pests dataset. O ViT demonstrou desempenho superior, alcançando uma precisão de 0.9574 sem aumento de dados. Embora o ViT não tenha se beneficiado do aumento de dados, essas técnicas se mostraram essenciais para melhorar o desempenho do AlexNet, ResNet-50 e EfficientNet. Nossos resultados ressaltam o potencial dos métodos de deep learning para a classificação de pragas, oferecendo ferramentas valiosas para ajudar os profissionais a manter a qualidade e o valor das culturas.

Palavras-Chave: Classificação de Pragas — Deep Learning — Aumento de Dados — Otimização de Hiperparâmetros

¹ Institute of Exact and Technological Sciences, Federal University of Viçosa (UFV), Brazil

*Corresponding author: joaofmari@gmail.com

DOI: <http://dx.doi.org/10.22456/2175-2745.143520> • Received: 24/10/2024 • Accepted: 02/12/2024

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. 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 learn-

ing (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 CNNs, 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 is an important task in agriculture because it directly affects crop yield and quality. However, it is challenging due to the wide variety of pest species, which differ based on region and crop type. Many pests change significantly throughout their life cycle, with stages such as eggs, larvae, pupae, and adults looking very different. Additionally, some pests hide within crops and can only be identified by the damage they cause. These challenges are further complicated by the limited availability of labeled datasets that cover this diversity. To address these issues, it is necessary to develop transfer learning techniques and deep learning models that can handle these variations effectively [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 2, we present the related works. In Section 3, we described the material and methods used, and in Section 4, the results are presented and discussed. Finally, we conclude the paper in Section 5.

2. 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. 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 Insect-Mamba 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 Insect-Mamba 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.

3. Material and methods

3.1 Dataset

We used the Agricultural Pests Dataset for our experiments. This dataset contains 5,494 images, divided into 12 classes: Ants, Bees, Beetles, Caterpillars, Earthworms, Earwigs,

Grasshoppers, Moths, Slugs, Snails, Wasps, and Weevils. The dataset is available on the Kaggle platform¹, 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 1 provides further details.

Table 1. 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

3.2 Architectures

We evaluated four architectures: AlexNet, ResNet-50, EfficientNet-B4, and Vision Transformer (ViT). AlexNet, a pioneering model in early CNN development, set the stage for modern deep learning techniques. ResNet-50 is recognized for its ability to scale deeper while effectively mitigating the vanishing gradient problem. EfficientNet-B4 balances accuracy and computational efficiency through its compound scaling approach. ViT, with its innovative self-attention mechanisms, offers a novel approach to image recognition. This diverse set of architectures allows us to examine how different architectural principles perform on the pest classification task.

AlexNet [13] 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 [14] 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 [15] 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 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) [16] 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.

3.3 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 224×224 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 224×224 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.

¹<https://www.kaggle.com/datasets/gauravduttakiit/agricultural-pests-dataset>



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

3.4 Experiment design

First, we randomly split 20% of the Agricultural Pest Dataset 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 3.2 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 3.3.

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.

3.5 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.

3.6 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:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

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

4. Results and Discussion

Table 2 shows the results of the hyperparameter optimization performed through grid search over the search space described in Section 3.4. The table presents the set of hyperparameter values that resulted in the best validation accuracy for each model.

Table 2. Optimized hyperparameter values for each architecture.

Architecture	BS	LR	Acc. Val.	Epochs
AlexNet	128	0.00001	0.8168	19
ResNet-50	16	0.00001	0.9352	5
EfficientNet	16	0.0001	0.9499	7
ViT b 16	32	0.00001	0.9613	17

Table 3 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.

We hypothesize that ViTs did not benefit from data augmentation in this task due to their self-attention mechanism, which effectively captures global features, thereby diminishing the necessity for additional transformations. In contrast, CNNs typically rely on augmentations such as random flips, rotations, and color jittering to enhance generalization, especially when dealing with datasets of limited variability [14, 16].

Tables 5 and 4 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 3. 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.

In Figure 3, we included the precision-recall curves for the ViT b 16 model evaluated on the test set under each data augmentation strategy. These results are presented specifically for the ViT b 16 model, which achieved the best performance across all metrics. Precision-recall curves are particularly useful for analyzing the trade-off between precision and recall for each class, providing information about how the model balances precision and recall. The area under the curve (AUC) for the precision-recall curves is displayed in parentheses in the figure. Across all data augmentation strategies, the ants and bees classes achieved the best performance, with an AUC of 1.00. In contrast, the beetle class proved to be the most challenging to classify, with AUC values ranging from 0.91 (DA-3) to 0.93 (No-DA and DA-2).

5. Conclusion

In this study, we compared four deep learning architectures for classifying pest images and assessed their performance with different data augmentation strategies. The models were fine-tuned using pre-trained models, and hyperparameter 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 focus on evaluating additional deep learning architectures and exploring novel training strategies, including advanced mix-based data augmentation techniques that more accurately reflect real-world conditions. Given the global diversity of insect pests and the significant visual differences they exhibit across various life stages, we plan to incorporate more complex datasets and leverage advanced transfer learning methods to improve model generalization and robustness.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. P. L. de O. Costa received a scholarship from PIBIC/CNPq.

Table 3. 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	<i>0.8144</i>	0.8107
	ResNet-50	0.9352	0.9374	0.9397	0.9363	0.9308	0.9263	<i>0.9381</i>	0.9227
	EfficientNet b4	0.9499	0.9534	0.9556	0.9488	0.9245	<i>0.9436</i>	0.9363	0.9190
	ViT b 16	0.9613	0.9590	0.9590	0.9511	0.9572	0.9454	0.9500	0.9472

Table 4. Experiment results with the models trained with different data augmentation strategies (weighted averaged).

		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	<i>0.8135</i>	0.8116	0.8110
	ResNet-50	0.9350	0.9377	0.9395	0.9367	0.9320	0.9270	<i>0.9383</i>	0.9226
	EfficientNet b4	0.9500	0.9539	0.9564	0.9505	0.9244	<i>0.9437</i>	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	<i>0.8144</i>	0.8107
	ResNet-50	0.9352	0.9374	0.9397	0.9363	0.9308	0.9263	<i>0.9381</i>	0.9227
	EfficientNet b4	0.9499	0.9534	0.9556	0.9488	0.9245	<i>0.9436</i>	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	<i>0.8103</i>	0.8054
	ResNet-50	0.9346	0.9363	0.9388	0.9358	0.9301	0.9255	<i>0.9375</i>	0.9216
	EfficientNet b4	0.9494	0.9522	0.9553	0.9478	0.9239	<i>0.9429</i>	0.9356	0.9179
	ViT b 16	0.9612	0.9586	0.9588	0.9506	0.9571	0.9452	0.9496	0.9471

Table 5. Experiment results with the models trained with different data augmentation strategies (macro averaged).

		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	<i>0.8063</i>	0.8049	0.8030
	ResNet-50	0.9314	0.9332	0.9356	0.9318	0.9290	0.9240	<i>0.9370</i>	0.9193
	EfficientNet b4	0.9473	0.9519	0.9536	0.9471	0.9215	<i>0.9424</i>	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	<i>0.8044</i>	0.8035
	ResNet-50	0.9301	0.9334	0.9343	0.9309	0.9278	0.9240	<i>0.9364</i>	0.9209
	EfficientNet b4	0.9450	0.9504	0.9527	0.9450	0.9224	<i>0.9405</i>	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	<i>0.8018</i>	0.7974
	ResNet-50	0.9302	0.9319	0.9341	0.9306	0.9269	0.9227	<i>0.9359</i>	0.9190
	EfficientNet b4	0.9455	0.9496	0.9523	0.9439	0.9214	0.9407	<i>0.9332</i>	0.9153
	ViT b 16	0.9578	0.9552	0.9554	0.9478	0.9553	0.9434	0.9468	0.9456

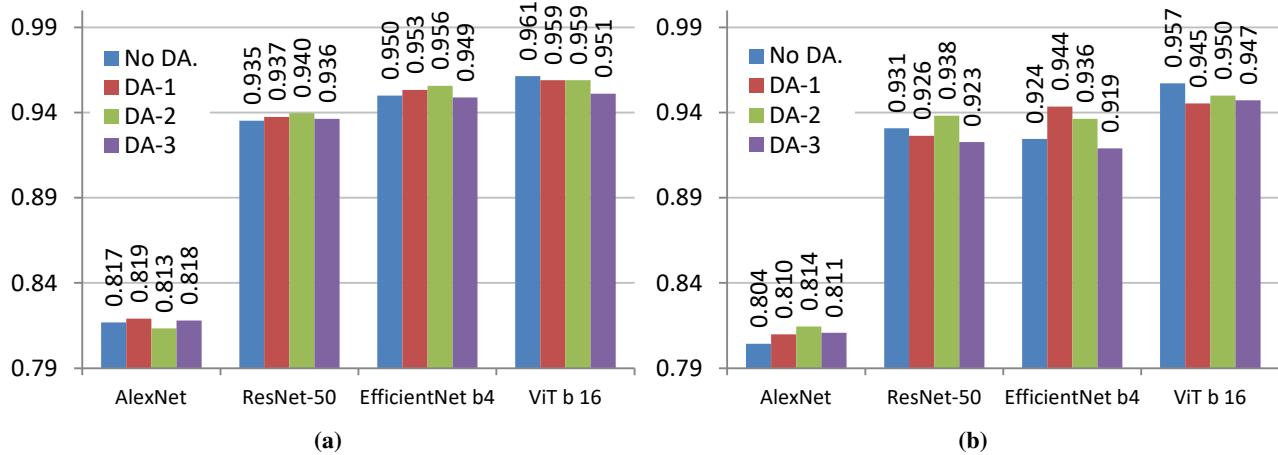


Figure 2. Charts with the accuracies obtained over the validation (a) and test (b) sets through the performed experiments.

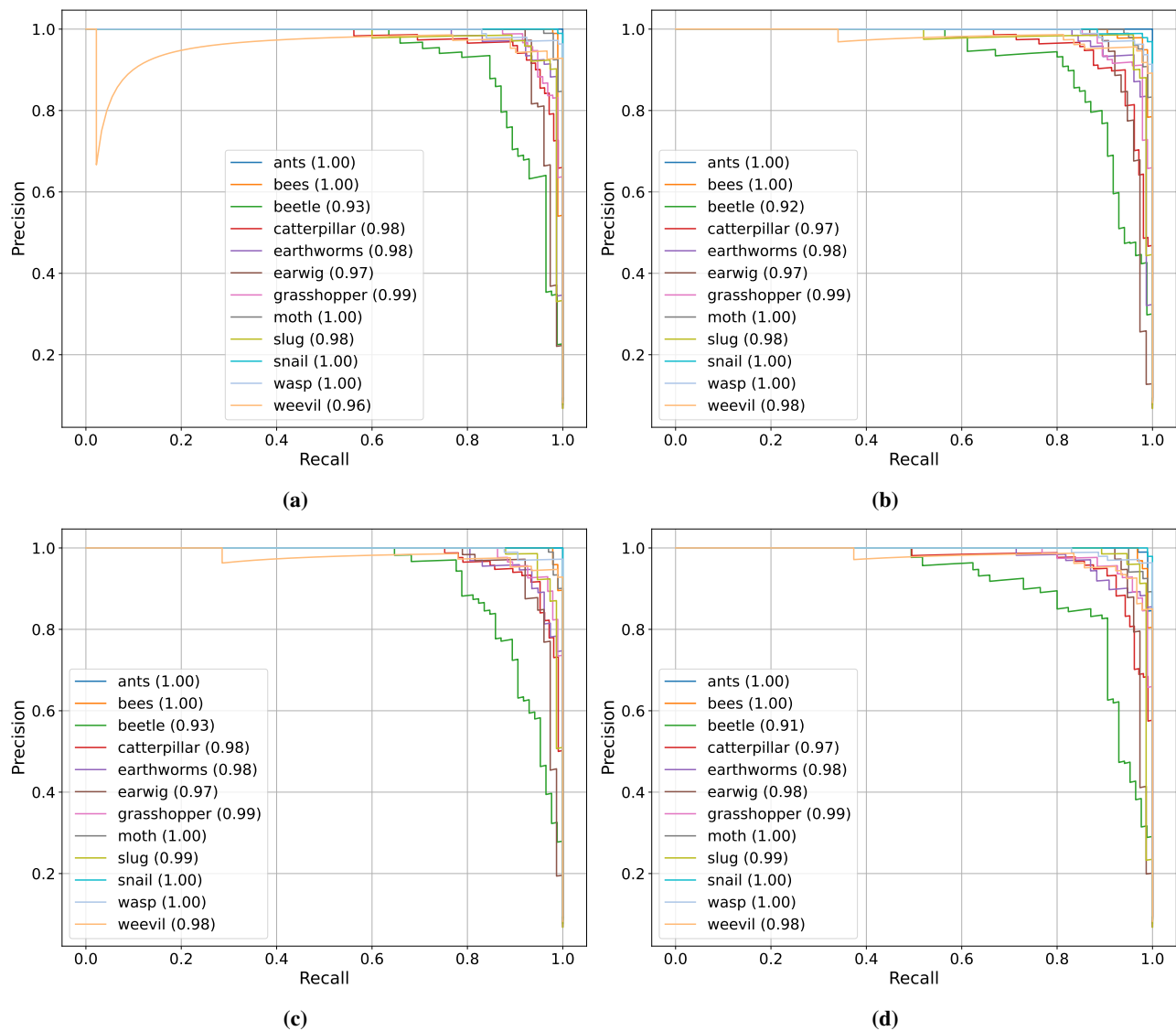


Figure 3. Precision-recall curves for the ViT b 16 model evaluated on the test set under different data augmentation strategies: (a) no data augmentation, (b) DA-1, (c) DA-2, and (d) DA-3.

Author contributions

Conceptualization: P.L.O.C., T.M.O.C., L.H.F.P.S., and J.F.M. Methodology: P.L.O.C., T.M.O.C., L.F.R.M., L.H.F.P.S., and J.F.M. Code Development: P.L.O.C., T.M.O.C., and J.F.M. Experiments: P.L.O.C., T.M.O.C., L.H.F.P.S., and J.F.M. Discussion of Results: P.L.O.C., T.M.O.C., L.F.R.M., L.H.F.P.S., and J.F.M. Writing - Original Draft: P.L.O.C., T.M.O.C., Writing - Review & Editing: P.L.O.C., T.M.O.C., L.F.R.M., L.H.F.P.S., and J.F.M. Supervision: L.F.R.M., L.H.F.P.S., and J.F.M.

References

- [1] PALEI, S. et al. Precision agriculture: MI and dl-based detection and classification of agricultural pests. In: IEEE. *2023 2nd International Conference on Ambient Intelligence in Health Care (ICAHC)*. 2023. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICAHC59020.2023.10431427>>.
- [2] ZHANG, W. et al. Research on the identification and detection of field pests in the complex background based on the rotation detection algorithm. *Frontiers in Plant Science*, Frontiers Media SA, v. 13, p. 1011499, 2022. Disponível em: <<https://doi.org/10.3389/fpls.2022.1011499>>.
- [3] NETO, R. G. P. et al. Enhancing green coffee quality assessment through deep learning. In: SBC. *Anais do XVIII Workshop de Visão Computacional*. 2023. p. 84–89. Disponível em: <<https://doi.org/10.5753/wvc.2023.27537>>.
- [4] ESGARIO, J. G. et al. An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning. *Information Processing in Agriculture*, Elsevier, v. 9, n. 1, p. 38–47, 2022. Disponível em: <<https://doi.org/10.1016/j.inpa.2021.01.004>>.
- [5] MOTA, G. S. de L. et al. Classifying pests in crop images using deep learning. In: SBC. *Anais do XVIII Workshop de Visão Computacional*. 2023. p. 42–47. Disponível em: <<https://doi.org/10.5753/wvc.2023.27530>>.
- [6] ULLAH, N. et al. An efficient approach for crops pests recognition and classification based on novel deepestnet deep learning model. *IEEE Access*, IEEE, v. 10, p. 73019–73032, 2022. Disponível em: <<https://doi.org/10.1109/ACCESS.2022.3189676>>.
- [7] KHANRAMAKI, M.; ASLI-ARDEH, E. A.; KOZEGAR, E. Citrus pests classification using an ensemble of deep learning models. *Computers and Electronics in Agriculture*, Elsevier, v. 186, p. 106192, 2021. Disponível em: <<https://doi.org/10.1016/j.compag.2021.106192>>.
- [8] WANG, Q. et al. Insectmamba: Insect pest classification with state space model. *arXiv preprint arXiv:2404.03611*, 2024. Disponível em: <<https://doi.org/10.48550/arXiv.2404.03611>>.
- [9] UTTARWAR, M. et al. An novel deep learning model for detection of agricultural pests and plant leaf diseases. In: IEEE. *2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. 2023. p. 1–6. Disponível em: <<https://doi.org/10.1109/CSDE59766.2023.10487697>>.
- [10] HUANG, Q. Comparison of deep transfer learning models for pest image classification in agriculture. *Highlights in Science, Engineering and Technology*, v. 94, p. 9–16, 2024. Disponível em: <<https://doi.org/10.54097/kxbxjn03>>.
- [11] BHATT, A. et al. Deep transfer learning based light-weight agricultural pest classification model using convolutional neural networks. In: IEEE. *2024 International Conference on Inventive Computation Technologies (ICICT)*. 2024. p. 175–180. Disponível em: <<https://doi.org/10.1109/ICICT60155.2024.10544420>>.
- [12] ABDULRAZZAQ, A. R.; TÜRK BEN, A. K.; KURNAZ, S. Detection and classification of agricultural pest and vermin's using full-convolutional neural network. In: IEEE. *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*. 2023. p. 1–6. Disponível em: <<https://doi.org/10.1109/ISAS60782.2023.10391393>>.
- [13] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012>.
- [14] HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [s.n.], 2016. p. 770–778. Disponível em: <<https://doi.org/10.48550/arXiv.1512.03385>>.
- [15] TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. 2019. p. 6105–6114. Disponível em: <<https://doi.org/10.48550/arXiv.1905.11946>>.
- [16] DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. Disponível em: <<https://doi.org/10.48550/arXiv.2010.11929>>.