

Evaluating Preprocessing and Data Augmentation for Soybean Leaf Pest Classification with Deep Learning

Avaliação de Pré-processamento e Aumento de Dados para Classificação de Pragas em Folhas de Soja com Deep Learning

Ana Luiza dos Santos Borges¹, Larissa Ferreira Rodrigues Moreira¹,
Leandro Henrique Furtado Pinto Silva¹, João Fernando Mari¹

Abstract: Soybean is one of the most important crops in the world, and its productivity is often reduced by pests that attack the leaves. Detecting these problems early is important for effective crop management. In this work, we evaluate three deep learning models, ResNet-50, ViT-B16, and ConvNeXt-Tiny, for the classification of soybean leaves as healthy or infested by Caterpillar or *Diabrotica speciosa*. The experiments were carried out using the public "Images of Soybean Leaves" dataset collected in real field conditions. We tested two preprocessing strategies (Resize and Random Resized Crop) and three levels of data augmentation (none, mild, strong). Results show that ConvNeXt-Tiny obtained the most stable performance, with its best results under Random Resized Crop and strong augmentation (accuracy of 0.9594, precision of 0.9555, and F1-score of 0.9576). ResNet-50 also performed well with cropping strategies, while only ViT-B16 benefited from simple Resize preprocessing. Macro and weighted metrics showed that the models kept good performance even with class imbalance. These results confirm the potential of deep learning for automatic pest detection in soybean leaves. Future work includes testing with larger datasets and using more advanced augmentation methods to improve robustness in real field conditions.

Keywords: soybean — pest classification — deep learning — preprocessing — data augmentation

Resumo: A soja é uma das culturas mais importantes do mundo, e sua produtividade é frequentemente reduzida por pragas que atacam as folhas. Detectar esses problemas de forma precoce é importante para o manejo eficiente da lavoura. Neste trabalho, avaliamos três modelos de deep learning, ResNet-50, ViT-B16 e ConvNeXt-Tiny, para a classificação de folhas de soja como saudáveis, infestadas por lagartas ou por *Diabrotica speciosa*. Os experimentos foram realizados com o conjunto de dados público "Images of Soybean Leaves", coletado em condições reais de campo. Foram testadas duas estratégias de pré-processamento (Resize e Random Resized Crop) e três níveis de aumento de dados (nenhum, leve e forte). Os resultados mostram que o ConvNeXt-Tiny apresentou o desempenho mais estável, com seus melhores valores no cenário com Recorte Aleatório e Redimensionamento e com aumento de dados forte (acurácia de 0,9594, precisão de 0,9555 e F1-score de 0,9576). O ResNet-50 também apresentou bom desempenho com estratégias de Recorte Aleatório e Redimensionamento, enquanto apenas o ViT-B16 se beneficiou do pré-processamento simples com Redimensionamento. As métricas macro e ponderadas mostraram que os modelos mantiveram bom desempenho mesmo com desbalanceamento de classes. Esses resultados confirmam o potencial do deep learning para a detecção automática de pragas em folhas de soja. Trabalhos futuros incluem testar conjuntos de dados maiores e aplicar métodos mais avançados de aumento de dados para aumentar a robustez em condições reais de campo.

Palavras-Chave: soja — classificação de pragas — deep learning — pré-processamento — aumento de dados

¹ Instituto de Ciências Exatas e Tecnológicas, Universidade Federal de Viçosa (UFV), Brazil

*Corresponding author: joaofmari@gmail.com

DOI: <http://dx.doi.org/10.22456/2175-2745.150920> • Received: 14/10/2025 • Accepted: 28/10/2025

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

1. Introduction

Soybeans are one of the most important crops in the world and hold a prominent position in Brazil, which is among the leading global producers and exporters of soybeans. Its cultivation is fundamental to the Brazilian economy, being widely used in the production of vegetable oil, animal feed, processed foods, and, more recently, biofuels [1]. In addition to their nutritional value and benefits to human health, soybean cultivation drives complex production chains and has a direct impact on the country's trade balance [2].

Brazil has vast areas of soybean cultivation, often operating under monoculture and conservation agriculture systems, the latter characterized by the use of practices that minimize soil and water loss [3]. However, these practices also increase the vulnerability of crops to pests and diseases, which can lead to significant losses throughout the production cycle [4].

Therefore, the use of technologies capable of quickly identifying signs of infestation in plantations is essential to ensure proper management and preserve agricultural yields. In this context, precision agriculture has been gaining ground with the integration of computational technologies aimed at field monitoring and automation [5]. Among these technologies, computer vision stands out as a promising approach for early disease diagnosis and pest control [6]. In agricultural applications, this technology is already used for tasks such as weed identification, plant counting, fruit quality analysis, and leaf anomaly detection [7, 8].

In this work, we evaluate the impact of different deep learning architectures, preprocessing strategies, and data augmentation levels on the task of automatic pest classification in soybean leaves. The goal is to identify which combinations of techniques lead to more accurate and robust models for early detection of leaf infestations in real field conditions.

The paper is organized as follows. Section 2 presents related work. Section 3 describes the dataset, preprocessing, and data augmentation. Section 4 shows and discusses the results. Section 5 gives the conclusion and future work.

2. Related works

Karlekar and Seal [9] presented a system, called SoyNet, for automatic classification of soybean diseases from leaf images. SoyNet has two main modules: (i) one for extracting the leaf region from the complete image and (ii) another for classifying soybean leaf diseases. The authors evaluated the approach using the Image Database of Plant Disease Symptoms (PDDb), which contains 16 classes of soybean leaf diseases. In terms of precision, recall, and F1-score, SoyNet achieved the best results compared to three other methods based on manual classification and nine other methods based on deep learning, demonstrating the robust capabilities of SoyNet.

Gui et al. [10] present a new classification method (CNN-SVM) for early detection of soybean leaf diseases using hyperspectral imaging. The authors present a combined convolutional neural network and support vector machine model to

rank disease severity at three levels. To evaluate the method, the authors demonstrated that CNN-SVM achieves an accuracy of 94.17% and outperforms standalone CNN models. Furthermore, CNN-SVM proved more robust for smaller sample sizes, maintaining accuracy above 90%.

Pereira et al. [11] proposed a new deep learning-based approach to identify and classify adult whiteflies and five related life stages (four instars and exuviae) in soybean leaf images. The detection algorithm is based on YOLOv4, with improvements in data augmentation, image mosaicking, and merging of hypothetical object categories. The authors evaluated the approach on an annotated dataset of 121 soybean leaf images, each containing 632 whitefly objects annotated into six different classes. The results demonstrated the viability of the strategy in terms of F1-score, especially when compared to YOLOv4 in its traditional configuration.

Zhu et al. [12] present an approach, CBF-YOLO, for detecting common soybean pests in complex environments. The main novelties of this approach include: (i) a CSE-ELAN module that improves feature extraction in spatial and channel dimensions to avoid missing pest features; (ii) a Bi-PAN structure that incorporates superficial features from the network to generate information with greater detail and pest localization; and (iii) an FFE module that eliminates conflicting information during feature fusion, improving network performance. The results demonstrated that the combined use of these modules led to a considerable improvement in network performance and mAP values, achieving 86.9% on the test set and 81.6% in real field scenes, outperforming other traditional detection models.

Kaur and Shalu [13] present a hybrid convolutional neural network (CNN-LSTM) with a long short-term memory (LSTM) model for feature extraction and classification of soybean leaf diseases. The authors used a spotted hyena optimizer (SHO) to iteratively adjust the weights and hyperparameters of the CNN-LSTM, exploring solutions until the best was found. The proposed hybrid approach demonstrated robustness, outperforming previous models and hyperparameter optimization studies in terms of accuracy, precision, and recall, while running faster.

Deep learning has been increasingly applied to precision agriculture, including disease and pest classification, showing strong potential for crop monitoring and early intervention. While many studies focus on detecting pest effects through leaf damage, others address direct pest classification in crop images. Mota et al. [14] conducted a comprehensive comparative analysis of multiple deep learning architectures, including CNNs and Vision Transformers, on customized versions of the IP102 dataset, showing the effectiveness of ViT for direct pest classification tasks. Similarly, Costa et al. [15] evaluated several architectures on the Agricultural Pests dataset, highlighting the impact of model choice and data augmentation. These works highlight the importance of architecture and training strategies for pest recognition, complementing our study, which evaluates how preprocessing and augmenta-

tion strategies influence the performance in soybean leaf pest classification.

3. Material and Methods

3.1 Dataset

We used the "Images of Soybean Leaves"¹[16] dataset, which contains soybean leaf images collected under real field conditions. Table 1 presents the distribution of images across the three classes in the dataset. The proportion of images per class reveals a moderately imbalanced scenario, with the *Caterpillar* class representing the majority (51.6%), followed by *Diabrotica speciosa* (34.4%) and *Healthy* samples (14.0%). The dataset was partitioned into training, validation, and test sets following a stratified split of 60%/20%/20%, ensuring that the original class proportions were preserved across all subsets. This strategy helps maintain the representativeness of each class during both model training and evaluation.

Table 1. Number of images per class and dataset split.

Class	Images	Images (%)	Train	Val.	Test
Caterpillar	3309	51.6%	1985	662	662
<i>Diabrotica speciosa</i>	2205	34.4%	1323	441	441
Healthy	896	14.0%	538	179	179
Total	6410	100.0%	3846	1282	1282

Figure 1 shows representative examples from the three dataset classes. Each row corresponds to five randomly selected images: (a) *Caterpillar*, (b) *Diabrotica speciosa*, and (c) *Healthy*. The samples illustrate the high intra-class variability and the visual similarities across classes, which are crucial for assessing model robustness and prediction performance.

3.2 Deep Learning Architectures

ResNet-50 is a CNN that introduces residual connections to enable stable training of very deep models [17]. Each residual block computes a nonlinear transformation $f(x)$ of the input x and adds it back to the original input, as shown in Equation 1:

$$H(x) = f(x) + x \quad (1)$$

where x is the input, $f(x)$ represents the block operations, and $H(x)$ is the output. This skip-connection mechanism mitigates vanishing gradients and allows effective optimization of very deep networks.

The Vision Transformer (ViT-B16) applies the Transformer architecture, originally designed for natural language processing, to image classification tasks [18]. An input image is divided into patches of size 16×16 , which are linearly projected into embeddings and combined with positional encodings. These embeddings are then processed by multi-head self-attention and feed-forward layers, enabling the model to capture global relationships across the image.

ConvNeXt-Tiny is a modern convolutional architecture that revisits the ResNet design with modifications inspired by Transformers [19]. Key changes include the use of Layer Normalization, GELU activations, inverted bottlenecks, depth-wise convolutions, and an updated downsampling strategy. These design choices improve accuracy and efficiency, enabling ConvNeXt to achieve performance comparable to vision Transformers while maintaining the advantages of CNNs.

3.3 Preprocessing and Data Augmentation Strategies

To assess the influence of preprocessing and data augmentation on classification performance, we adopted two distinct image preprocessing pipelines: Resize and Random Resized Crop. Each pipeline was evaluated using three levels of augmentation intensity, *none*, *mild*, and *strong*, while keeping the augmentation procedures consistent across both pipelines. This design allowed a controlled comparison between preprocessing effects and augmentation levels.

In the Resize pipeline, all input images were scaled to a fixed resolution of 224×224 pixels. This strategy standardizes image dimensions to match the input size expected by models pretrained on ImageNet. However, it may introduce geometric distortions when the original aspect ratio of the image is not preserved [20]. In contrast, the Random Resized Crop pipeline selects a random region within the original image, resizes it to 224×224 , and passes it to the network. This introduces spatial variability and simulates zooming or cropping effects, often improving generalization by emphasizing local regions of the image [17].

The augmentation levels were designed to introduce increasing amounts of variability. In the *none* (or baseline) condition, only preprocessing and normalization using the dataset's mean and standard deviation were applied. The *mild* augmentation level added random horizontal flipping (with 50% probability) and random rotation up to $\pm 15^\circ$, which are common strategies to increase model robustness to pose and orientation changes [21]. The *strong* augmentation configuration extended the mild one by introducing color jittering to simulate variations in brightness, contrast, saturation, and hue. This additional photometric distortion encourages the model to learn color-invariant features and better handle diverse lighting conditions [22].

Although the augmentation operations were kept identical for both pipelines, the distinct preprocessing approaches introduce different spatial features. Resize preserves the global layout of the image, whereas Random Resized Crop inherently disrupts it by focusing on subregions. Therefore, we evaluate both pipelines independently, allowing us to distinguish the individual contributions of augmentation intensity and preprocessing strategy to the final model performance.

It is important to note that, for both validation and testing, we consistently applied the Resize preprocessing without any data augmentation. This decision ensures determinism and fairness during evaluation. Unlike training, where each

¹<"<https://data.mendeley.com/datasets/bycbh73438/1>">



Figure 1. Sample images from the Soybean Leaf dataset. Rows (a), (b), and (c) display five randomly selected images from each class: *Caterpillar*, *Diabrotica speciosa*, and *Healthy* leaves, respectively. These samples illustrate the visual variability within and across classes.

image is presented to the model multiple times with different transformations, test samples are processed only once. Consequently, applying Random Resized Crop during evaluation would introduce unwanted randomness and inconsistent cropping, which is inappropriate for reliable performance assessment.

3.4 Training Procedure

All models were implemented in PyTorch using pretrained models available in torchvision and fine-tuned using the cross-entropy loss function, suitable for multi-class classification. The optimization was performed using the AdamW optimizer with an initial learning rate of 0.0001. A mini-batch size of 16 was adopted in all experiments.

To adjust the learning rate during training, we adopted a plateau-based scheduling strategy, which monitors the validation loss and reduces the learning rate when no improvement is observed, with a patience of 10 epochs and a reduction factor of 0.1. This approach helps to refine the learning process by allowing longer convergence when the model reaches performance plateaus.

We trained the models for a maximum of 400 epochs. However, to prevent overfitting and reduce unnecessary computation, we applied early stopping based on the validation

loss. Specifically, training was halted if the validation loss did not improve for 21 consecutive epochs (patience = 21). The model corresponding to the lowest validation loss was saved to disk and later used for both evaluation and inference purposes. During training, we recorded the training and validation loss, as well as accuracy at each epoch, in order to monitor model performance and convergence behavior.

3.5 Model Evaluation

The performance of the models was assessed on the respective test sets using the following classification metrics: overall accuracy, precision, recall, and the F1-score. To ensure a fair comparison across imbalanced datasets, both macro-averaged and weighted-averaged versions of the metrics were reported. All measures are derived from the entries of the confusion matrix, namely true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

Overall accuracy quantifies the global proportion of correctly classified instances and is defined as Equation 2:

$$\text{Accuracy} = \frac{\sum_{c=1}^C \text{TP}_c}{\sum_{c=1}^C (\text{TP}_c + \text{FP}_c + \text{FN}_c)}, \quad (2)$$

where C denotes the number of classes.

Table 2. Accuracy for different models, resizing strategies, and data augmentation levels.

Preprocessing	Resize			Random Resized Crop		
	None	Mild	Strong	None	Mild	Strong
ResNet-50	0.9259	0.9555	0.9477	0.9392	0.9626	0.9485
ViT-B16	0.9033	0.9548	0.9462	0.9540	0.9509	0.9524
ConvNext-Tiny	0.9321	0.9462	0.9563	0.9501	0.9548	0.9594

Table 3. Macro and weighted metrics Precision for different models, resizing strategies, and data augmentation levels.

Reduction	Macro						Weighted					
	Resize			Random Resized Crop			Resize			Random Resized Crop		
Data Aug.	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong
ResNet-50	0.9253	0.9447	0.9483	0.9368	0.9550	0.9372	0.9265	0.9564	0.9478	0.9413	0.9629	0.9489
ViT-B16	0.8871	0.9485	0.9370	0.9455	0.9432	0.9417	0.9056	0.9549	0.9462	0.9546	0.9510	0.9530
ConvNext-Tiny	0.9339	0.9440	0.9496	0.9526	0.9534	0.9555	0.9330	0.9463	0.9566	0.9502	0.9549	0.9596

Table 4. Macro and weighted metrics Recall for different models, resizing strategies, and data augmentation levels.

Reduction	Macro						Weighted					
	Resize			Random Resized Crop			Resize			Random Resized Crop		
Data Aug.	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong
ResNet-50	0.9088	0.9564	0.9392	0.9376	0.9609	0.9472	0.9259	0.9555	0.9477	0.9392	0.9626	0.9485
ViT-B16	0.8969	0.9513	0.9372	0.9513	0.9479	0.9484	0.9033	0.9548	0.9462	0.9540	0.9509	0.9524
ConvNext-Tiny	0.9249	0.9428	0.9602	0.9439	0.9496	0.9600	0.9321	0.9462	0.9563	0.9501	0.9548	0.9594

Table 5. Macro and weighted metrics F1-score for different models, resizing strategies, and data augmentation levels.

Reduction	Macro						Weighted					
	Resize			Random Resized Crop			Resize			Random Resized Crop		
Data Aug.	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong	None	Mild	Strong
ResNet-50	0.9164	0.9500	0.9436	0.9358	0.9576	0.9420	0.9255	0.9556	0.9476	0.9388	0.9625	0.9486
ViT-B16	0.8907	0.9499	0.9371	0.9480	0.9454	0.9447	0.9035	0.9548	0.9462	0.9540	0.9509	0.9525
ConvNext-Tiny	0.9290	0.9433	0.9545	0.9481	0.9514	0.9576	0.9323	0.9461	0.9563	0.9500	0.9547	0.9594

Precision and recall, and F1-score were computed for each class c as follows:

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

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}. \quad (4)$$

$$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \quad (5)$$

To summarize results across classes, we report the macro average and the weighted average, which accounts for class imbalance:

$$\text{F1}_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C \text{F1}_c; \quad (6)$$

$$\text{F1}_{\text{weighted}} = \frac{\sum_{c=1}^C n_c \cdot \text{F1}_c}{\sum_{c=1}^C n_c}. \quad (7)$$

3.6 Computational Environment

All experiments were conducted on a workstation featuring an Intel Core i5 processor (3.0 GHz), 32 GB of RAM, and an NVIDIA GeForce GTX 1080 Ti GPU with 11 GB of memory. The development environment was based on Python 3.12 and PyTorch 2.7.1 with CUDA 12.6 support. Model implementation and data augmentation were carried out using torchvision

0.22.1. Additional tools for data preprocessing, visualization, and metric computation included scikit-learn 1.7, scikit-image 0.25.2, Pillow 11.0, and Matplotlib 3.10.3.

4. Results and discussion

Table 2 presents the overall accuracy achieved by the three evaluated architectures (ResNet-50, ViT-B16, and ConvNeXt-Tiny) under different preprocessing strategies (Resize and Random Resized Crop) and data augmentation levels (None, Mild, and Strong). Table 3 reports the precision values, while Table 4 shows the corresponding recall values for the same experimental configurations. Finally, Table 5 presents the F1-scores, the harmonic mean of precision and recall that offers a balanced view of both metrics, particularly useful in scenarios with class imbalance. For precision, recall, and F1-score, both macro and weighted reductions are presented to capture overall performance and class imbalance effects.

Here, we analyze the results presented in Tables 2–5 according to three perspectives: resizing strategy, data augmentation, and the distinction between macro and weighted evaluation.

4.1 Resize vs. Random Resized Crop

The effect of the resizing strategy varied across architectures, but the best results for all models were obtained with Random Resized Crop. ConvNeXt-Tiny delivered strong and consistent performance under both strategies, achieving the highest macro-level scores with Random Resized Crop combined with strong augmentation (accuracy of 0.9594, precision of 0.9555, and F1-score of 0.9576). The only exception was recall, where ResNet-50 with mild augmentation outperformed ConvNeXt-Tiny, also under Random Resized Crop.

ViT-B16 showed weaker results compared to the other models and, unlike them, performed better with the simple Resize preprocessing than with Random Resized Crop. This behavior may be explained by ViT-B16's use of fixed patch embeddings and global attention, which makes it more sensitive to spatial changes introduced by cropping, unlike CNN-based models that benefit from local receptive fields and translation invariance.

Overall, ConvNeXt-Tiny proved less sensitive to the resizing choice, while ResNet-50 and ViT-B16 benefited more from the variability introduced by cropping.

4.2 Impact of Data Augmentation

The intensity of data augmentation also had a clear effect. For ResNet-50 and ViT-B16, mild augmentation consistently improved performance, leading to their best results across several metrics. For ConvNeXt-Tiny, strong augmentation produced the highest scores in all metrics except recall, suggesting that this model is more tolerant to aggressive transformations, especially when combined with Random Resized Crops.

4.3 Macro vs. Weighted Evaluation

The comparison between macro and weighted metrics shows that results are generally consistent across models, with weighted scores usually higher than the macro ones, especially in the best-performing configurations. This behavior is expected, since weighted metrics emphasize frequent classes, while macro metrics balance contributions from all classes. The relatively small gap between them indicates that, although class imbalance influences the results, the models are still able to maintain good performance across less frequent classes.

Figure 2 presents the training and validation curves for loss (top) and accuracy (bottom) of the best-performing model from each architecture, selected based on accuracy. ResNet-50 (Figure 2(a)) converges within approximately 25 epochs, ViT-B16 (Figure 2(b)) shows gradual improvement up to 40 epochs, and ConvNeXt-Tiny (Figure 2(c)) reaches stable performance in fewer than 10 epochs. In all cases, the final model was determined by the early stopping criterion with a patience parameter of 21.

Figure 3 presents the Precision–Recall curves for the best-performing model of each architecture, selected according to accuracy. ResNet-50 (Figure 3(a)) and ConvNeXt-Tiny (Figure 3(c)) achieve almost identical curves with high precision across the recall range, while ViT-B16 (Figure 3(b)) shows similarly consistent behavior with a slightly smaller area under the curve for the Healthy class. In all cases, the curves remain close to the upper-right corner, indicating robust performance across classes.

5. Conclusion

In this work, we evaluated three deep learning architectures, ResNet-50, ViT-B16, and ConvNeXt-Tiny, for automatic pest classification in soybean leaves. The experiments considered different preprocessing strategies and levels of data augmentation, and results were analyzed with both macro and weighted evaluation metrics. ConvNeXt-Tiny achieved the most stable performance, with strong augmentation and Random Resized Crop leading to the best overall results. ResNet-50 and ViT-B16 also benefited from data augmentation, though their performance was more dependent on the preprocessing strategy. The comparison between macro and weighted metrics showed that, although class imbalance had some influence, the models were able to maintain good accuracy across all classes.

As future work, we plan to expand the study with larger and more diverse datasets, apply advanced augmentation techniques such as CutMix and MixUp. Testing on more varied conditions will be an important step to improve robustness and practical use. These directions aim to strengthen the application of computer vision for early pest detection, supporting more effective and sustainable soybean crop management.

Acknowledgements

We would like to thank CAPES, CNPq, and FAPEMIG for the financial support. This study was financed in part by

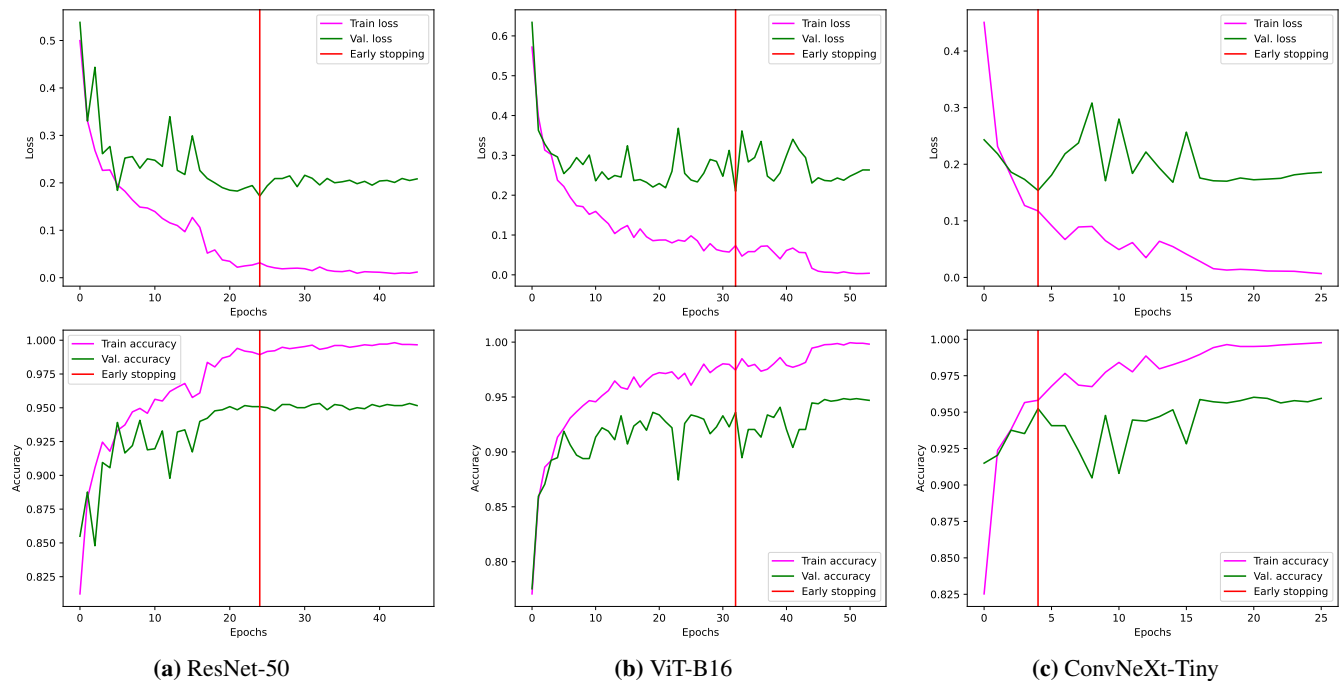


Figure 2. Training and validation curves for loss (top row) and accuracy (bottom row) of the best-performing model from each architecture, selected based on classification accuracy. (a) ResNet-50 trained with Random Resized Crop and strong data augmentation; (b) ViT-B16 trained with Resize and mild data augmentation; (c) ConvNeXt-Tiny trained with Random Resized Crop and strong data augmentation.

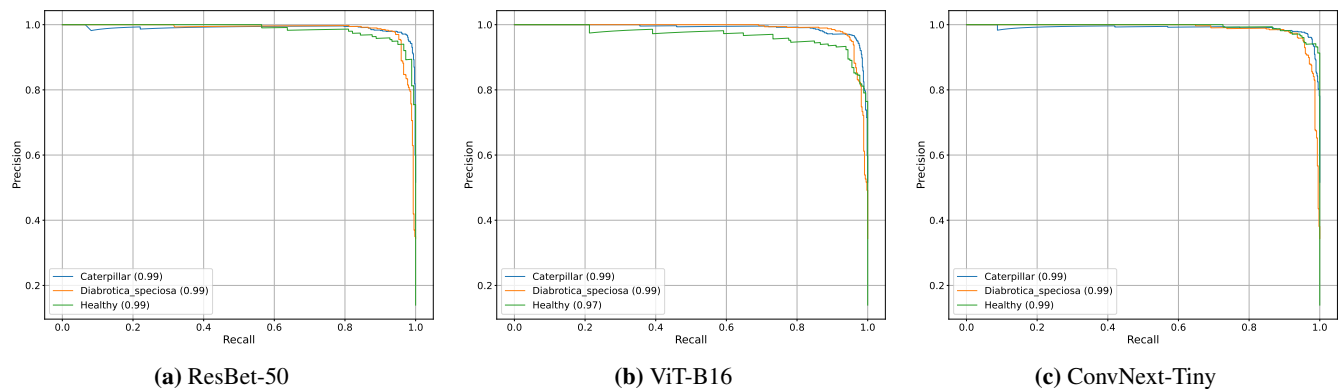


Figure 3. Precision–Recall curves for the best-performing model of each architecture, selected based on classification accuracy. (a) ResNet-50 trained with Random Resized Crop and strong data augmentation; (b) ViT-B16 trained with Resize and mild data augmentation; (c) ConvNeXt-Tiny trained with Random Resized Crop and strong data augmentation.

the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

L.F.R.M., L.H.F.P.S., and J.F.M.

Author contributions

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

References

- [1] SILVA, M. V. da et al. Generating synthetic multispectral images using neural style transfer: A study with application in channel alignment. *Computers and Electronics in Agriculture*, Elsevier, v. 206, p. 107668, 2023.
- [2] KARMAKAR, P. et al. Crop monitoring by multimodal remote sensing: A review. *Remote Sensing Applications: Society and Environment*, Elsevier, v. 33, p. 101093, 2024.

- [3] PANIZZZI, A. R. History and contemporary perspectives of the integrated pest management of soybean in brazil. *Neotropical entomology*, Springer, v. 42, n. 2, p. 119–127, 2013.
- [4] ALVES, B. J.; BODDEY, R. M.; URQUIAGA, S. The success of bnf in soybean in brazil. *Plant and soil*, Springer, v. 252, n. 1, p. 1–9, 2003.
- [5] SISHODIA, R. P.; RAY, R. L.; SINGH, S. K. Applications of remote sensing in precision agriculture: A review. *Remote sensing*, MDPI, v. 12, n. 19, p. 3136, 2020.
- [6] CISTERNAS, I. et al. Systematic literature review of implementations of precision agriculture. *Computers and Electronics in Agriculture*, Elsevier, v. 176, p. 105626, 2020.
- [7] SOUSSI, A. et al. Smart sensors and smart data for precision agriculture: a review. *Sensors*, MDPI, v. 24, n. 8, p. 2647, 2024.
- [8] RAM, B. G. et al. A systematic review of hyperspectral imaging in precision agriculture: Analysis of its current state and future prospects. *Computers and Electronics in Agriculture*, Elsevier, v. 222, p. 109037, 2024.
- [9] KARLEKAR, A.; SEAL, A. SoyNet: Soybean leaf diseases classification. *Computers and Electronics in Agriculture*, Elsevier, v. 172, p. 105342, 2020.
- [10] GUI, J. et al. Grading method of soybean mosaic disease based on hyperspectral imaging technology. *Information Processing in Agriculture*, Elsevier, v. 8, n. 3, p. 380–385, 2021.
- [11] PEREIRA, R. de C. et al. Detection and classification of whiteflies and development stages on soybean leaves images using an improved deep learning strategy. *Computers and Electronics in Agriculture*, Elsevier, v. 199, p. 107132, 2022.
- [12] ZHU, L. et al. Research on cbf-yolo detection model for common soybean pests in complex environment. *Computers and Electronics in Agriculture*, Elsevier, v. 216, p. 108515, 2024.
- [13] KAUR, J.; SHALU. An optimized hybrid neural network framework for prognosis of leaf diseases in soybean crops using spotted hyena optimizer. *International Journal of Information Technology*, Springer, p. 1–7, 2025.
- [14] 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>>.
- [15] COSTA, P. L. de O. et al. Classification of agricultural pests through digital images using deep learning. *Revista de Informática Teórica e Aplicada*, v. 32, n. 1, p. 18–25, 2025. Disponível em: <<https://doi.org/10.22456/2175-2745.143520>>.
- [16] MIGNONI, M. E. *Images of Soybean Leaves*. Mendeley Data, 2021. Disponível em: <<https://doi.org/10.17632/bycbh73438.1>>.
- [17] HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.
- [18] DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2021.
- [19] LIU, Z. et al. A convnet for the 2020s. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2022. p. 11976–11986.
- [20] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. v. 25.
- [21] SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, Springer, v. 6, n. 1, p. 1–48, 2019.
- [22] SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1–9.