

Automation of Coffee Biotic Stresses Classification Using Convolutional Neural Networks and Image Preprocessing Techniques

Automatização da Classificação de Estresses Bióticos no Café Usando Redes Neurais Convolucionais e Técnicas de Pré-processamento de Imagens

Daniel Imai Yamakawa¹, Ana Claudia Patrocínio², Pedro Moises de Sousa^{1*}, Pedro Henrique Aparecido Damaso de Melo^{1**}

Abstract: Coffee is one of the most consumed beverages worldwide, with Brazil leading exports in 2024. However, production is threatened by biotic stresses such as cercospora, phoma, rust, and leaf miner, which reduce grain quality and cause major losses. Disease identification is usually performed manually by specialists, a process that is slow, limited, and error-prone. This study proposes automating classification through Convolutional Neural Networks (CNNs) combined with various image processing techniques. The JMunBEN and JMunBEN2 datasets were used for this task, together with image preprocessing techniques such as cropping, augmentation, Gaussian, Clahe, Graythresh and Wavelet. Three CNN architectures—ResNet50, MobileNetV2, and AlexNet—were tested for multiclass classification, trained for 100 epochs with Adam optimizer and CrossEntropy loss. The article proposed wavelet filter yielded the best results, with ResNet achieving 99.07% accuracy, confirming CNNs with image pre-processing techniques to be efficient tools for rapid, reliable detection of coffee biotic stresses.

Keywords: coffee — neural networks — artificial intelligence — wavelet

Resumo: O café é uma das bebidas mais consumidas no mundo, com o Brasil liderando as exportações em 2024. A produção, porém, enfrenta desafios devido a estresses bióticos como cercospora, phoma, ferrugem e bicho-mineiro, que reduzem a qualidade dos grãos e causam perdas significativas. A identificação dessas doenças é feita manualmente por especialistas, tornando o processo lento e sujeito a falhas. Este trabalho propõe o uso de Redes Neurais Convolucionais (CNNs) juntamente com técnicas de processamento de imagens para automatizar a classificação. Foram utilizados os bancos JMunBEN e JMunBEN2 e técnicas de pré-processamento como cropping, data augmentation, Gaussian, Clahe, Graythresh e Wavelet. Três modelos foram avaliados (ResNet50, MobileNetV2 e AlexNet), sendo ResNet50 com filtro de wavelet proposto o mais eficiente, alcançando 99,07% de acurácia.

Palavras-Chave: café — redes neurais — inteligência artificial — wavelet

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

²Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia (UFU), Brazil

*Corresponding authors: ¹{daniel.yamakawa, pedromoises, pedro.henrique.melo}@ufv.br, ²{ana.patrocinio}@ufu.br

DOI: <http://dx.doi.org/10.22456/2175-2745.150942> • Received: 15/10/2025 • Accepted: 03/11/2025

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

1. Introduction

Coffee is one of the most popular beverages worldwide. In 2024, Brazil stood out as its largest exporter, with 50.44 million bags of coffee exported, with a new historical record of both performance export volume and foreign exchange revenue, which reached US\$12.51 billion in this period, highlighting the importance in coffee production [1]. However, several challenges in coffee cultivation affect its overall pro-

duction and the quality of the beans, such as biotic stresses. Among these stresses are Cercospora, phoma leaf spot, coffee leaf rust, and the coffee leaf miner [2], which manifest on the leaves and cause significant losses for coffee farmers.

The classification of these biotic stresses is performed manually and in person, which can pose challenges for timely execution since it depends on the availability of a specialist visiting the coffee plantation. It is not always possible to rely on a professional being available, and furthermore, manual

classification is subject to human error [2].

Technology is rapidly evolving and has begun to be utilized in the automation of classification processes [3]. Examples of applications in agriculture include a corn disease detection system [4], soybean disease identification [5], and cotton yield prediction [6], demonstrating the efficiency and effectiveness in the use of these technological approaches. This innovation is also used in classifying coffee biotic stresses, achieving reliable and rapid results, which is essential for addressing the time consuming and human error issue. This enables faster decision-making for purchasing and applying treatments to combat these biotic stresses, along with other necessary actions [7].

Convolutional Neural Networks (CNNs), a branch of artificial intelligence, are widely used for image classification in agriculture, delivering highly accurate results, as seen in the study by [8], which achieved high accuracy in classifying coffee biotic stresses such as leaf miner, leaf rust, cercospora, and phoma leaf spot. Some studies, such as [9], take the process even further by enabling image classification using smartphones, emphasizing the approach's potential. In [2], custom architecture was developed specifically for classifying coffee diseases, obtaining promising results and the study by [8] evaluates the severity levels of the diseases.

In order to solve the cited problems, this work aims to develop and study approaches using image pre-processing techniques together with CNNs. The goal is to create tools that can recognize biotic stresses presented in leaves, such as coffee leaf miner, Phoma disease, coffee rust, and Cercospora leaf spot. This will aid in the classification and control of these abnormalities, using various methods to increase the accuracy rate. A proposed wavelet image pre-processing method was also tested and evaluated for comparison with the other preprocessing methods.

2. Theoretical Framework

To identify coffee leaf diseases, it is necessary to absorb diverse characteristics. Deep learning and machine learning are areas of Artificial Intelligence and fit the task very well.

2.1 Convolutional Neural Networks (CNNs)

The main idea of CNN is to obtain features from an input (usually an image) and the deeper the layer, the more complex the details extracted. It is important to point out that this process is not cheap in terms of computational cost, so when a large database is used, a GPU is used [10]. There are several CNN models, the most famous for identifying agrarian diseases are AlexNet [11], created in 2012, being revolutionary for its time with good and consistent results, VGGNet [12] with an error rate of 7.32%, one of the best presented in 2014, GoogLeNet [13] which won the ILSVRC 2015 with an error rate of 6.7%, and ResNet [14] proposed by *Microsoft Research Asia* and has one of the best histories for recognizing diseases in the agricultural and medical fields [10].

2.1.1 Architecture of a CNN

CNNs consist of a sequence of different types of layers for different tasks. A CNN usually has a convolutional layer, a function activation layer, a pooling layer, a fully connected layer and a dropout layer. All these layers together form a functional CNN architecture, which can also include other optional layers such as normalization and dropout to solve the problem of *overfitting*, a common problem that occurs when the model gets too used to a situation and cannot generalize [10].

2.1.2 Convolutional Layers

Convolutional layers are the fundamental basis of a CNN that uses convolutional operations. Their parameters are filters called *kernels*. The main task of these layers is to detect features in local regions and map their appearance to a feature map, which is obtained in each filter in the layer by repeating the same process a few times until the image is complete [10].

Stacking multiple convolutional layers allows the network to learn hierarchical features: lower layers capture simple patterns like edges, while deeper layers represent more complex structures. Often, convolutional layers are followed by pooling layers, which reduce spatial dimensions and focus on the most important features [10].

2.1.3 Pooling Layer

In CNNs, the next layer after convolutional and activation is the *pooling* layer. This layer takes all the maps from the convolutional layer and summarizes them. There are some *pooling* techniques, and one of them is *max-pooling*, where the largest possible value is output. For example, if the input is a 2 x 2 region, *max-pooling* will maximize the four values. These operations eliminate data that are not necessary for the functioning of the CNN and preserve any crucial features in smaller resolutions [10].

2.1.4 Fully Connected Layers

The final layer that joins the features with their hidden states in the first fully connected layer. It is possible to use more than one fully connected layer to improve computational performance. Fully connected layers have most of the parameters. For example, if each of two fully connected layers has 4096 hidden units, the connections between them would have more than 16 million weights. The nature of fully connected layers can depend directly on the application used, a fully connected layer of a classification application would differ from one used in segmentation, for example [6].

2.2 Wavelet in 2 dimensions

Wavelets have been shown as powerful foundations for signal processing and image analysis at more than one resolution. A transformation function joins all these wavelets to form a function or image. The two-dimensional wavelet transformation is frequently used in image preprocessing and compression. It is necessary to separate the one-dimensional wavelet function into three coefficients, which are horizontal, vertical and diagonal [15].

$$\begin{aligned}
 LL(x,y) &= \sum_m \sum_n f(m,n) h(x-2m) h(y-2n), \\
 LH(x,y) &= \sum_m \sum_n f(m,n) h(x-2m) g(y-2n), \\
 HL(x,y) &= \sum_m \sum_n f(m,n) g(x-2m) h(y-2n), \\
 HH(x,y) &= \sum_m \sum_n f(m,n) g(x-2m) g(y-2n),
 \end{aligned} \tag{1}$$

These coefficients measure different variations, first applied in the first dimension to all horizontal, vertical and finally the diagonals [15]. The original image is divided into four sub-images: LL (*Low-Low*): The low-frequency part in the horizontal and vertical direction. LH (*Low-High*): The low-frequency part in the horizontal direction and high-frequency in the vertical direction. HL (*High-Low*): The high-frequency part in the horizontal direction and low-frequency in the vertical direction. HH (*High-High*): The high-frequency part in both directions. After the decomposition into components, the original image is reconstructed.

3. Related Work

The study [9] aims to facilitate the recognition of coffee leaf diseases using CNNs through a mobile application. The BRACOL database was used, which is a public database with a total of 1747 images of Arabica coffee and was divided into classes for healthy leaves, leaf miner, rust, phoma, and brown eye spot, with the addition of some images from another study and techniques such as data collection, labelling, and data augmentation to make the image database more robust. The architectures used in this work were UNet [16] and PSPNet [17] for segmentation, and AlexNet, GoogLeNet, VGG16, and ResNet50 [14] for symptom classification. The best result was obtained by ResNet50 with an accuracy of 97.07%, a precision of 96.85%, and a recall of 96.69%.

In [18], the study is centered on diagnosing coffee diseases, specifically Brown Eye Spot and Rust, by comparing two different approaches: one studying leaf texture and another using a CNN. A private database from the Federal Institute of Southern Minas Gerais - Muzambinho-MG was used, with 750 sub-images of each class after data augmentation was applied. For the first approach, Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) were used, while the CNN used was the AlexNet architecture, trained with 20 epochs and a learning rate of 0.001. In the experiment using the texture attribute, a sensitivity of 0.933 was obtained in the best experiment, while the experiment using AlexNet obtained a sensitivity of 0.980, showing that both methods are reliable to assist coffee workers, with AlexNet performing slightly better [10].

In the [8] study, coffee diseases are classified along with their severity, studying leaf miner, rust, brown eye spot, and phoma disease. The database is composed of 1747 images of

Arabica coffee, with 372 images showing more than one disease, and another 2147 images cropped to show only the symptoms of each disease, with the addition of 575 images from another database for the symptoms. The architectures used were adapted for multi-task learning, which means performing two tasks at the same time: disease classification and disease severity. Using the AlexNet, GoogLeNet, VGG16, ResNet50, and MobileNetV2 [19] architectures, we obtained a disease classification accuracy of 96.58%, 96.82%, 95.60%, 97.07%, and 96.33% respectively, using the symptom database. With the multi-task approach using the normal database, we obtained 86.9%, 82.94%, 86.51%, and 84.13% for the severity of each disease [10].

4. Methods

Certain steps are necessary for the system to recognize biotic stresses in coffee, as shown in Figure 1 and 2.

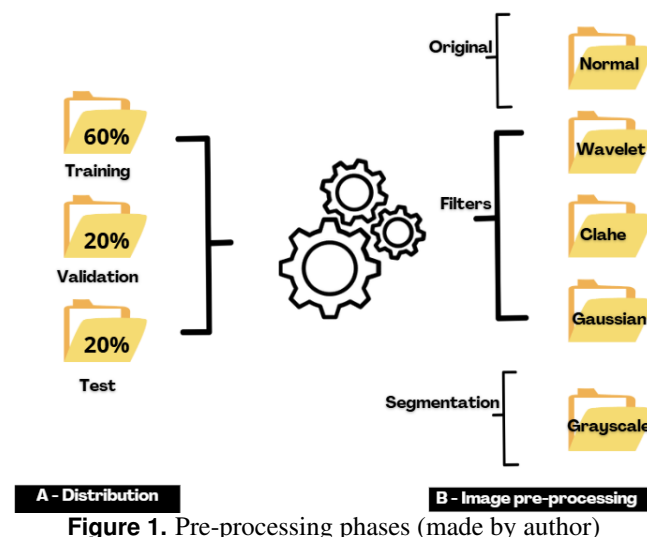


Figure 1. Pre-processing phases (made by author)

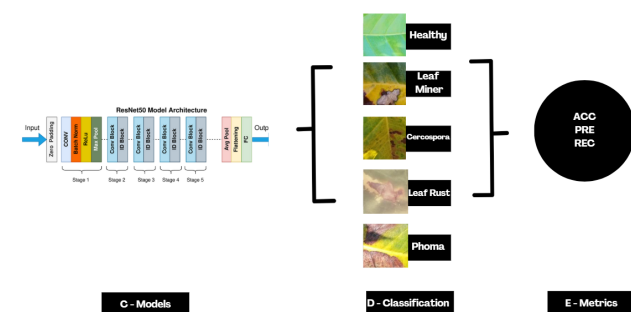


Figure 2. Image classification (made by author)

In figure 2, the images after the filters were applied were used for training, validation and test for the models explained in section 4.3, and were assigned one of the classes in the image, and as a form of validation and evaluation metrics were utilized as explained in further detail in the manuscript.

4.1 Class Distribution

For this research, public databases for Arabica coffee images JMunBEN and JMunBEN2 [20] were used for testing, training and validation. It is an image bank specially designed for research related to disease recognition and classification of Arabica coffee leaves, containing a total of 58,555 images divided into 5 classes (phoma disease, cercospora, rust, leaf miner, healthy leaves) of 128x128 pixels and JPG files. 1000 of each class images were selected for the experiments.

The image database was divided into three parts: Training containing 60% of the images, testing containing 20% and validation 20%. This division is essential for the system to work, as shown in Fig. 3.

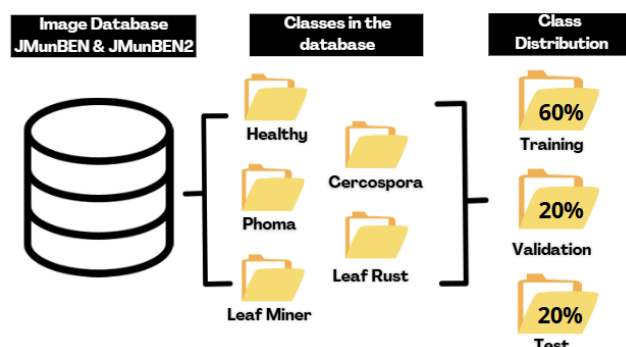


Figure 3. Class distribution (made by author)

4.2 Image Pre-processing:

Techniques to improve the quality and quantity of the images were used on the image database itself, and filters containing different techniques to improve the result were applied separately.

4.2.1 Cropping

Each image in this image database was checked to ensure they were all in the same format. A cropping tool was used to remove unnecessary areas of the image to reduce the training time in image processing [20].

4.2.2 Data Augmentation

Data augmentation was performed to increase the quantity of available images, by rotating and flipping them in a way to give emphasis to the areas of interest. Rotating was done at an angle of 180° horizontally and vertically [20].

4.2.3 Gaussian

The Gaussian filter is a function used to "blur" images in order to remove distortions and details from the image [21].

It applies an "average" of the central neighboring pixels of the image, providing a smoothing effect on the image and preserving the edges better than other similar filters [21].

It is possible to change the values of how much the details will be reduced, and in this case, a sigma of 0.5, the standard deviation of gaussian distribution, using a 2-D gaussian smoothing kernel was applied to the database [21].

4.2.4 CLAHE

CLAHE (Contrast Limited Adaptive Histogram Equalization) is a variant of AHE (Adaptive Histogram Equalization). The contrast enhancement is done by the neighborhood of a pixel and transformed by the slope of the transformation function [22].

CLAHE works on small areas of an image instead of the complete image, so the technique is applied within each block. To minimize the excessive amplification of distortions, it is possible to define a contrast limit. The applied limit was a value of 0.02 [22].

4.2.5 Graythresh

Graythresh is a function used to calculate the ideal binarization of an image using the Otsu method [23].

The Otsu method is a core mathematical engine used for graythresh, implemented to automatically and optimally determine the best threshold value that separates the pixels into two distinct classes (foreground and background) by maximizing their difference. Then, a binary image with those two classes of pixels is generated using that method.

In this work, Graythresh was applied to segment and binarize images in the dataset, facilitating the extraction of meaningful features and identification of areas of interest. Binarization is particularly useful in applications like object detection, shape analysis, and pattern recognition, where distinguishing objects from the background is crucial. By converting images into binary form, subsequent computational tasks such as edge detection or morphological analysis become more efficient and accurate by making the images simpler to process.

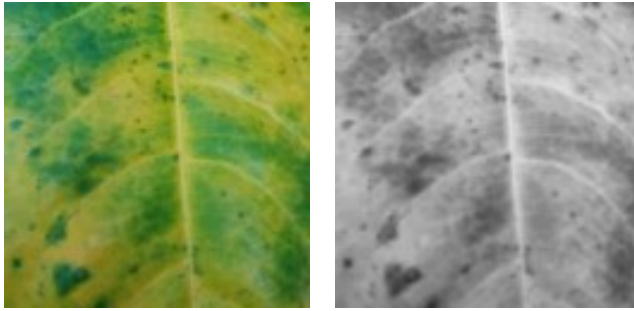
4.2.6 Wavelet

The wavelet image filter was used to decompose the image into signals of different scales, serving for noise removal, image compression, and changing its resolution [15]. Unlike traditional Fourier transforms, which only provide frequency information, wavelet transforms preserve both frequency and spatial information, making them particularly suitable for analyzing localized image features using the 2-dimensional wavelet filter. The wavelet filter enhances important features for classification, making it a valuable preprocessing technique in deep learning pipelines [15].

4.2.7 Wavelet proposed image pre-processing technique

A wavelet filter was applied with Gaussian filter, focused on noise reduction with the Discrete Wavelet Transform method. Each image was applied a Graythresh function, and then the Gaussian function was applied and the Inverse Wavelet Transform method was used to construct the final image.

There were tests performed in order to use the best wavelet family for this kind of research. Wavelets families such as bior1.1, Daubechies wavelet(db3 - db45), Symlet wavelet (sym1 - sym45), Fejér-Korovkin wavelet (fk1 - fk22) and finally Meyer wavelet (dmey) families were tested. As a form of evaluation, this article used PSNR to measure performance



(a) Leaves figure from database (JMUBEn2) (b) Proposed pre-processing applied (made by author)

Figure 4. Comparison of two leaves figures, on the left the normal image and on the right the proposed image pre-processing applied

of each family.

$$PSNR = 10 \log \left(\frac{MAX(I_{out})^2}{MSE} \right) = 20 \log \left(\frac{MAX(I_{out})}{MSE} \right)$$

In this equation, MAX is the maximum value a pixel can assume in the image and MSE is the mean squared error (MSE) [24]

While fk22 obtained an average of 46.56 PSNR, the chosen family was dmey with 46.61 PSNR.

4.3 Models

The ResNet50 is a highly popular, 50-layer deep neural network primarily used for computer vision tasks like image classification, object detection, and face recognition. It's main purpose is to be an excellent feature extractor, meaning it learns to identify complex patterns, shapes, and textures in images, which it then uses to make predictions. The ResNet50 was applied by modifying the final layer for multiclass classification, with the Adam optimizer, CrossEntropyLoss criterion and a learning rate of 0.001, with some experiments being pre-trained and others not, with 100 epochs trained. It obtains great numbers in this type of task [14].

MobileNetV2 is a highly efficient convolutional neural network (CNN) architecture designed primarily for mobile and embedded vision applications. It's main purpose is to perform complex computer vision tasks with high accuracy while maintaining low computational cost, small model size, and low latency. With the ReLu6 activation function, it was used modifying the last layer for multiclass classification and Adam optimizer, CrossEntropy criterion with a learning rate of 0.001, pre-trained with it's experiments run for 100 trained epochs, obtaining good results [19].

AlexNet is a pioneering Convolutional Neural Network (CNN) architecture that primarily excels at image classification and marked a pivotal moment in the resurgence of deep learning. It was the first widely recognized application of deep CNNs for large-scale visual recognition, winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 by a significant margin. The AlexNet model was used

by modifying one of it's layers for multiclass classification with Adam optimizer and CrossEntropy criterion, with a learning rate of 0.001 and the experiment using 100 epochs, being a reliable model in various tasks [11].

4.4 Metrics used

Different metrics are applied to the results to get a better dimension of the performance of each model in each image filter, which helps to evaluate and interpret the results, facilitating the comparison between the different methods.

The following components are used to calculate different metrics

Accuracy: Proportion of correct predictions made by a model in relation to the total number of predictions. It is calculated as the number of correct predictions divided by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Precision (PRE): Proportion of correct positive predictions made by the model in relation to the total number of positive predictions made. It is used to measure the exactness of positive predictions.

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

Recall (REC): Proportion of true positives that the model was able to identify in relation to the total number of actual positive cases. It measures the model's ability to find all positive instances.

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

4.5 Experimental Environment

The experiments were conducted on Python 3.10 with the Pytorch version 2.5.1 for it's expanded use in works of the same area. It provides great support for Neural Networks and libraries to aid it's use. Tests were carried out on a Asus TUF Dash F15 laptop with a 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz (3.30 GHz) processor, 24,0 GB RAM and RTX3060 Mobile.

5. Results and Discussion

A 100-epoch test was conducted using the ResNet50, MobileNetV2, and AlexNet models, with the results presented in Table 1.

The application of image filters showed promising results. On average, the experiments with the Gaussian filter yielded higher numbers, while the Clahe and Graythresh segmentation filters also showed good results, as seen in Table 2.

Figure 5a shows the accuracy for MobileNetV2 graph, while Figure 5c shows for AlexNet and Figure 5e shows

Table 1. Neural network architectures and applied filters

Architecture	Applied Filter	ACC (%)	PRE (%)	REC (%)
ResNet50	Normal	97,87%	98,00%	97,00%
	Clahe	97,54%	97,00%	97,00%
	Gaussian	98,01%	98,00%	98,00%
	Graythresh	96,68%	97,00%	96,00%
	Wavelet	93,01%	93,00%	93,00%
MobileNetV2	Dmey	99,07%	99,00%	99,00%
	Normal	97,93%	98,00%	98,00%
	Clahe	97,41%	97,00%	97,00%
	Gaussian	98,01%	98,00%	98,00%
	Graythresh	96,68%	97,00%	97,00%
AlexNet	Wavelet	94,64%	95,00%	95,00%
	Dmey	98,81%	99,00%	98,00%
	Normal	97,56%	98,00%	98,00%
	Clahe	97,43%	98,00%	98,00%
	Gaussian	98,25%	98,00%	98,00%
	Graythresh	96,29%	96,00%	96,00%
	Wavelet	77,43%	78,00%	77,00%
	Dmey	99,03%	99,00%	98,00%

Table 2. Average loss with each filter applied for all architectures overall per image filter

Image Filters	Average Acc	Average Loss
Normal	97.78%	0.1082
Clahe	97.46%	0.1196
Gaussian	98.09%	0.1066
Graythresh	96.55%	0.1580
Wavelet	88.36%	0.3877
Dmey	98.97%	0.0821

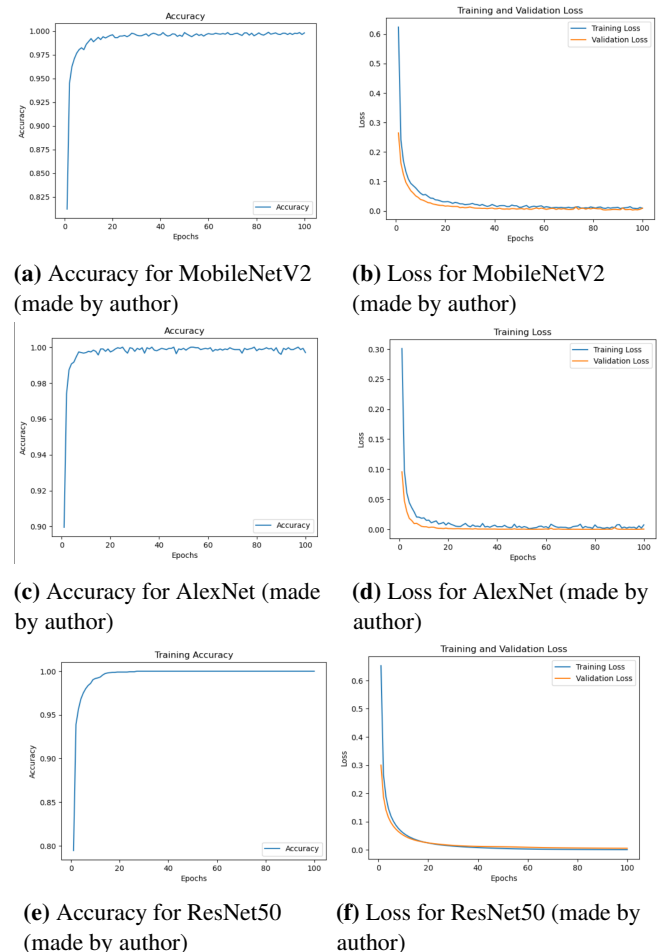
for ResNet50 graphs for the proposed image pre-processing wavelet dmey family proposed method. MobileNetV2 like AlexNet had a more inconsistent showing, while ResNet50 showed very stable results.

In Figure 5b a loss graph is shown for MobileNetV2, and in Figure 5d one for AlexNet and finally in Figure 5f the loss graph for ResNet50

This work evaluated the performance of the ResNet50, AlexNet, and MobileNetV2 architectures in this task, using the JMunBEN and JMunBEN2 databases [20]. The Gaussian filter obtained a very good performance, with the second lowest loss and second accuracy compared to the other filters in general, while the image with the Wavelet technique resulted in the lowest accuracy, but still yielded excellent performance on the ResNet50 and MobileNetV2 networks. The average of all filters was 98.09% using the Gaussian filter, 97.78% with the normal database, 97.46% with the Clahe filter, 96.55% with the Graythresh segmentation method, and finally 88.36% with the Wavelet filter, which was significantly lowered due to its performance on the AlexNet network.

It is shown that the proposed dmey image pre-processing method had better results, having the highest average with 98.97% and the lowest average loss, 0,0821. Dmey wavelet performance also showed great compability with JMunBEN and JMunBEN2 databases.

The obtained precision and recall results are similar to the accuracy results, with the Gaussian filter showing the best

**Figure 5.** Comparison of accuracy across different architectures (made by author)

performance among the other filters, as presented in Table 2. The wavelet filter ended up presenting the worst result in both precision and recall, due to the greater difficulty the models had in classifying images with this filter applied.

Table 3 compares the results of the work in [18] and [9] with the model of the proposed paper applied with the filters. Better accuracies were presented for the ResNet50, MobileNetV2, and AlexNet architectures, with the best accuracy result of the work [18] for the ResNet50 architecture being 95.63%, compared to the proposed work with the wavelet dmey and with the Clahe and Gaussian filters, which obtained 99.07%, 97.54%, and 98.01% respectively. The MobileNetV2 architecture from the work [18] obtained 94.84% compared to the results of the proposed work, which were 98.81% on the dmey wavelet, 97.41% on the Clahe filter, and 98.01% on the Gaussian filter. The AlexNet model proposed by the work [18] reached 92.46%, while the proposed work reached 99.03% on the dmey wavelet, 97.43% on the Clahe filter, and 98.25% on the Gaussian filter. The work [9] obtained 98.00% sensitivity, and the proposed work obtained 98.25% accuracy, 98.00% precision, and 98.00% recall with the AlexNet model.

Table 3. Comparison of metrics obtained by the three best proposed filters with state-of-the-art methodology.

Architecture	Applied Filter	My model			[17], 2020			[9], 2019
		ACC (%)	PRE (%)	REC (%)	ACC (%)	PRE (%)	REC (%)	Sensitivity (%)
ResNet50	Dmey	99,07%	99,00%	99,00%	95,63%	94,12%	92,70%	-
	Clahe	97,54%	97,00%	97,00%				
	Gaussian	98,01%	98,00%	98,00%				
MobileNetV2	Dmey	98,81%	99,00%	98,00%	94,84%	93,24%	91,79%	-
	Clahe	97,41%	97,00%	97,00%				
	Gaussian	98,01%	98,00%	98,00%				
AlexNet	Dmey	99,03%	99,00%	98,00%	92,46%	89,37%	90,02%	98,00%
	Clahe	97,43%	98,00%	98,00%				
	Gaussian	98,25%	98,00%	98,00%				

6. Limitations of the Study

This paper has some limitations that have to be acknowledged. The study used the JMUBen and JMUBen2 databases however more accurate results would be obtained with a bigger range of different databases all together to form a single one. In addition to that, the current classification system only works with one biotic stress, not being able to classify more than one in the same image. This study does not provide an easier way to use it's services, thus lacking a port to use this technology easily. The current implementations are a prototype and can be worked on and improved in future research with different databases, a more accessible way to run the classification program, and possibly a multi classification system to identify multiple biotic stresses at the same time without having any drawbacks.

Another limitation in this study is the lack of testing different parameter configurations for the CLAHE and Gaussian filters, which could potentially lead to more optimized preprocessing and improved classification performance.

7. Conclusion

This study successfully demonstrated the effectiveness of using convolutional neural networks (CNNs) in conjunction with image pre-processing techniques for the automated classification of biotic stresses in coffee leaves. Using JMunBEN and JMunBEN2 public databases, this work evaluated the performance of three CNN architectures being ResNet50, MobileNetV2, and AlexNet, using different image filters, including a novel wavelet-based pre-processing method. The results consistently that these technological approaches can achieve high accuracy, offering a reliable alternative to traditional, manual inspection methods that are often time-consuming and prone to human error and the impacts of each image filters.

We conclude that the filters vary in results, with Dmey and Gaussian obtaining the highest averages on each architecture being 98.87% and 98.09% respectfully. The average loss in all experiments were on low, having at most 0.3877 in normal wavelet filter and 0.0821 in the dmey proposed filter, being the the lowest of all experiments. It is shown that in

the task of classifying the biotic stresses of rust, leaf miner, phoma spot, and brown eye spot to assist farmers in planting Arabica coffee this work is highly efficient and reliable, with results close to 100% accuracy. We obtained better results than previous works with the use of image filters, demonstrating the difference in metrics when using them.

The proposed image pre-processing wavelet technique shows that the family dmey together with the custom denoising method can be used for future works in hopes of achieving greater results.

While the study presents significant results, it is important to acknowledge it's limitations. The research used a specific set of public databases, and future work could benefit from a larger, more diverse database to improve the model's generalization capabilities. The current system is designed to classify a single disease per image, while a multi-label classification system capable of identifying multiple biotic stresses simultaneously would be more desirable. Finally, developing a user-friendly application (e.g., for smartphones) would make this technology more accessible to coffee farmers, enabling faster and more informed decision-making regarding crop health.

Overall, this research confirms that CNNs, when combined with advanced pre-processing techniques, are a powerful tool for automating disease detection in agriculture. The promising results pave the way for future developments aimed at creating more robust, accessible, and versatile systems to support global coffee production.

Author Contributions

Daniel Imai Yamakawa: Conceptualization, Methodology, Software, Investigation, Formal analysis, Original Draft, Review & Editing. Ana Claudia Patrocínio: Conceptualization, Methodology, Validation, Resources, Supervision, Project administration, Review. Pedro Moises de Sousa: Conceptualization, Methodology, Validation, Resources, Supervision, Project administration, Review & Editing. Pedro Henrique Aparecido Damaso de Melo: Conceptualization, Methodology, Validation, Resources, Supervision, Project administration, Review & Editing.

References

- [1] CECAFE. *Conselho dos Exportadores de Café do Brasil*. 2024. Acesso em: 30 de agosto de 2025. Disponível em: http://www.consorciopesquisacafe.com.br/images/stories/noticias/2021/2024/Dezembro/CECAFE_Relatorio_Mensal_DEZEMBRO_2024.pdf.
- [2] RODRIGUES, F. et al. Desafios do manejo no controle de doenças do café - f i t o s s a n i d a d e. In: _____. [S.l.: s.n.], 2019. p. 52 – 54. ISBN 18066403.
- [3] SHARMA, R. Artificial intelligence in agriculture: A review. In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. [S.l.: s.n.], 2021. p. 937–942.
- [4] MISHRA, S.; SACHAN, R.; RAJPAL, D. Deep convolutional neural network based detection system for real-time corn plant disease recognition. *Procedia Computer Science*, v. 167, p. 2003–2010, 2020. ISSN 1877-0509. International Conference on Computational Intelligence and Data Science. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187705092030702X>.
- [5] BEVERS, N.; SIKORA, E. J.; HARDY, N. B. Soybean disease identification using original field images and transfer learning with convolutional neural networks. *Computers and Electronics in Agriculture*, v. 203, p. 107449, 2022. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169922007578>.
- [6] TEDESCO-OLIVEIRA, D. et al. Convolutional neural networks in predicting cotton yield from images of commercial fields. *Computers and Electronics in Agriculture*, v. 171, p. 105307, 2020. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169919319878>.
- [7] KRANZ, J. Measuring plant disease. In: _____. *Experimental Techniques in Plant Disease Epidemiology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. p. 35–50. ISBN 978-3-642-95534-1. Disponível em: https://doi.org/10.1007/978-3-642-95534-1_4.
- [8] ESGARIO, J. G.; KROHLING, R. A.; VENTURA, J. A. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*, v. 169, p. 105162, 2020. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169919313225>.
- [9] 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*, v. 9, n. 1, p. 38–47, 2022. ISSN 2214-3173. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214317321000044>.
- [10] WANI FAROOQ AHMAD BHAT, S. A. A. I. K. M. A. *Advances in Deep Learning*. Singapore: Springer Singapore, 2019.
- [11] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [12] SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.
- [13] SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 1–9.
- [14] HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.
- [15] ZHANG, D. Wavelet transform. In: _____. *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*. Cham: Springer International Publishing, 2019. p. 35–44. ISBN 978-3-030-17989-2. Disponível em: https://doi.org/10.1007/978-3-030-17989-2_3.
- [16] RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N. et al. (Ed.). *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015. p. 234–241. ISBN 978-3-319-24574-4.
- [17] ZHAO, H. et al. *Pyramid Scene Parsing Network*. 2017.
- [18] Boa Sorte, L. X. et al. Coffee leaf disease recognition based on deep learning and texture attributes. *Procedia Computer Science*, v. 159, p. 135–144, 2019. ISSN 1877-0509. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050919313468>.
- [19] SANDLER, M. et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019.
- [20] JEPKOECH, J. et al. Arabica coffee leaf images dataset for coffee leaf disease detection and classification. *Data in Brief*, v. 36, p. 107142, 2021. ISSN 2352-3409. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352340921004261>.
- [21] D’HAeyer, J. P. Gaussian filtering of images: A regularization approach. *Signal Processing*, v. 18, n. 2, p. 169–181, 1989. ISSN 0165-1684. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165168489900480>.
- [22] PIZER, S. et al. Contrast-limited adaptive histogram equalization: speed and effectiveness. In: *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*. [S.l.: s.n.], 1990. p. 337–345.

- [23] OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979.
- [24] PONTI, M. A.; COSTA, G. B. P. da. *Como funciona o Deep Learning*. 2018. Disponível em: <https://arxiv.org/abs/1806.07908>.