Faculty of Engineering Journal

Development of an detection model for brown spot disease in rice plants

Desarrollo de un modelo de detección de la enfermedad mancha parda en plantas de arroz

Nicolás Calderón, Dubán Estiben Cardozo Osorio.
Surcolombiana University
Avenida Pastrana Borrero, Carrera 1, 4100
Neiva, Colombia

KEYWORDS
English keywords
Machine Vision; Vision Transformers (ViT); Brown Spot; Rice Diseases; Image Segmentation.

ABSTRACT: The present project aims at developing a Machine Vision (ViT) model for the early detection of the "brown spot" disease in rice. The study addresses the problem of the decline in rice production in Huila due to factors such as climate change and the spread of diseases. "Brown spot", caused by the fungus Bipolaris oryzae, affects the photosynthesis of the plant.

The project aims to build a representative dataset, define an appropriate ViT model and optimize its accuracy through preprocessing and data augmentation. The methodology is divided into six phases: dataset creation, selection of preprocessing methods, architecture definition, training, testing and evaluation of the model. Advanced models such as Swin Transformer (v2) and DeiT are used to improve robustness in classification and segmentation.

1. Introduction

Rice is one of the most consumed cereals worldwide, being an important source of fiber, minerals and vitamins [1]. It was the third food with high consumption in Colombia in 2021 [2], with a production of approximately 3 million tons. However, rice production in Huila faces challenges due to water deficit, which has led to a decrease of 12,407 tons of production between 2016 and 2021. In recent decades, climate change has increased temperatures and altered rainfall patterns, affecting crop health by promoting the proliferation of pests and diseases [3].

One of the diseases that harms the rice plant is brown spot (Bipolaris oryzae), which occurs on the stem and leaf, causing lesions in the form of oval spots that become dark when extended, resulting in a significant reduction in chlorophyll concentrations and a deterioration of photosynthetic capacity [4].

Currently, there are different convolutional neural network (CNN) algorithms that are used to detect various diseases in rice plants. According to a recent study [5], these algorithms can reach 92.68% effectiveness in detecting diseases; However, these models often face limitations in terms of accuracy and segmentability capabilities. [6]

For this reason, this project implements a Transformers of Vision (ViT) model that focuses on increasing the accuracy of brown spot identification at an early stage in rice. [7]

2. Objectives

2.1. General objective

To develop a Transformers of Vision (ViT) model to improve the accuracy in the identification of brown spot in the rice plant.

2.2. Specific objectives

- Build a dataset that incorporates images of different rice crops with brown spot disease and healthy plants, ensuring an adequate representation of variations in growing conditions.

- Select image preprocessing methods that optimize the quality and representativeness of the dataset, including data augmentation techniques to improve the diversity and robustness of the image set.

- Define the specific architecture of the ViT model that will be implemented during training, adjusting the necessary parameters and components to maximize the accuracy in the detection of brown spot disease.

- Train an Artificial Vision model that allows detecting brown spot disease so that it can be accurate and reliable in recognition.

- Run test tests that evaluate the model's performance in detecting brown spot, using metrics such as accuracy, recovery, and F1 score to ensure its effectiveness.

- Evaluate the model in the detection of brown spot, analyzing the results obtained in the testing tests and adjusting the model as necessary to improve its performance.

3. Methodology

3.1. Phase 1. Creating the Dataset

3.1.1. Defining Classes

In this initial phase, the classes that will serve as a guide for the organization and labeling of the images were established. Each of these classes has been selected considering various conditions, such as:

- Brown spot: Images of leaves affected by brown spots, usually appearing as small circular or elliptical lesions, dark brown to reddish in color.
- Brown narrow spot: Images of leaves with narrow, brown spots, which usually appear aligned with the veins of the leaves, causing progressive damage.
- Pod blight: Images of leaves with irregular spots on the leaf sheaths, progressing from gray to dark brown spots and eventually affecting the leaves.
- Healthy leaf: Images of rice leaves in a completely healthy state, with no visible spots or signs of disease.
- Bacterial leaf blight: Images of rice leaves that show symptoms of bacterial blight, including watery lesions that eventually turn brown on the edge of the leaves, often with a wilted appearance.
- Piricularia (leaf burn): Images of leaves that exhibit symptoms of Pyricularia burn, with rounded or oval gray to brown spots, often expanding into severely affected areas.
- Leaf scald: Images of leaves with symptoms of scald, including elongated, whitish spots, surrounded by a dark brown halo.
- Piricularia on the neck: Images showing the attack on the top of the stem, where the panicle is located, causing a brown or black discoloration that can lead to the death of the plant in that area.
- Rice hyspa: Images that capture leaves affected by the rice hyspa insect, where frayed leaves are observed and perforated by the larvae that feed on the leaf tissue.
- Tungro: Images of rice leaves infected by the tungro virus, which are characterized by a yellowish or orange coloration, with delayed plant growth.

3.1.2. Image collection

After defining the classes, the collection of images from various sources will begin. Priority will be given to well-known websites such as Roboflow, Kaggle, and Google Images. These photographs will be taken of different rice leaves that the brown spot presents in order to obtain as much information as possible and thus ensure a good representativeness of the dataset.

3.1.3. Image conversion

At this stage, the conversion of the collected images to the .jpg format will be carried out using the Irfanview program. This process will ensure the uniformity of the format of all the images in the dataset, facilitating their handling and subsequent processing.

3.1.4. Balanceo del Dataset

A thorough review will then be performed to ensure that each of the pre-defined classes has the same number of images. In the event that any class has an insufficient number of images, examples will be searched or resampling techniques such as Downsampling will be applied to match the number of samples in all classes of the dataset.

3.2. Phase 2. Selection of processing methods

3.2.1. Data standardization

The images are then resized to a resolution of 224x224 pixels. This procedure ensures that the images have the same dimensions, simplifying post-processing and ensuring consistency in the input data for the model.

3.2.2. Data augmentation

After the standardization subphase is completed, data augmentation techniques will be applied to the dataset. This involves rotations in 8 directions: clockwise (45°, 90°, and 135°), counterclockwise (45°, 90°, and 135°), and inverted (up and down). In addition, the original images are converted to grayscale, helping to improve variability and the model's ability to generalize under different lighting and color conditions.

3.3. Phase 3. Definition of Deep Learning Architecture

3.3.1. Architecture Identification

First, the architectures that will be used as a basis for the research were identified. having as a selection models that allow working with datasets of 15,000 images, such as Vision Transformer (ViT), Swin Transformer (v2) and Data-efficient Image Transformer (DeiT)

3.3.2. Selection of architectures

Once the identification is complete, the architectures will be compared using the precision metric as the main criterion. This will allow the selection, in a well-founded way, of the three deep learning architectures that present the greatest value in this metric.

3.4. Phase 4. Training the model

3.4.1. Dataset Partitioning

After having chosen the deep learning architectures with which to work, the dataset will be divided into training, validation and test sets, following the following distribution:

Table 1. Dataset Distribution

Dataset Name	Percentage (%)
Training	70
Validation	20
Tests	10
Total	100

Source: Authors.

3.4.2. Training Setup

Next, the model's training hyperparameters, such as batch size and epochs, will be set.

3.4.3. Training Start

Once configured, training will begin using the Dataset prepared in the previous phases. During that process, each of the models will learn to recognize the patterns associated with brown spot disease, gradually adjusting their weights and parameters to minimize the loss function.

3.4.4. Training Results

Once the training is completed, the results obtained will be represented in various tables that summarize the performance of each model. These tables will contain relevant information such as batch size, epochs, and evaluation metrics, mainly accuracy.

3.5. Phase 5. Test Execution

3.5.1. Start of tests

Next, tests will be carried out using the test dataset, which represents 10% of the images in the Dataset. These will be carried out in Google Colab, using an Nvidia T4 GPU with 16 GB of VRAM, which will allow the performance of each of the models to be evaluated. It is important to note that in these tests the models with the best precision of each of the previously selected Deep Learning architectures will be used.

3.5.2. Test Results

Subsequently, the results derived from the tests will be shown, presenting specific examples of the predictions generated by each model. These results will be visualized through tables, which will incorporate performance metrics such as accuracy and response times, offering a detailed understanding of the performance of each model.

4. Results

4.1. Mathematical model

4.1.1 Windowed Self-Attention Calculation

For Swin Transformer, input is processed in local windows, where each window calculates its attention independently. This allows complexity to be maintained

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{dx}}\right)V$$

Equation 1. Calculation of the window of attention

Where:

- Q: Query array, represents the feature vector of each pixel in the window.
- K: Key matrix, which is used to compare the similarity between pixels within the same window.
- V: Value matrix, which contains the information of the pixels for attention.
- dx: The dimension of the key matrix, used to normalize the similarity score.
- softmax: Activation function that converts similarities into probabilities.

4.1.2. Window Shift

Window shifting allows the model to capture relationships between pixels that are not in the same original window.

$$Shifted\ Attention\ (q,k,v) = Attention(Q_{shift}K_{shift}V_{shift})$$

Equation 2. Window Scrolling

Where:

- $(Q_{shift}K_{shift}V_{shift})$: Displaced versions of the Q, K, and V matrices. Each window moves a predefined number of pixels (usually half a window), which allows information to be mixed between windows.

4.1.3 MLP (Multilayer Perceptron) Blocks

After the attention layer, each window passes through a multilayered perceptron network that helps transform the information in a non-linear manner.

$$MLP(x) = \sigma(W_2(GELU(W_1 x + b_1)) + b_2$$

Equation 3. MLP Blocks

Where:

- X: input from the self-care layer.
- $W_1 x + b_1$ weight matrices for the linear layers within the MLP.
- b_1: Bias vectors for each layer.
- GELU: Gaussian Error Linear Unit activation, used to improve performance in deep networks.

- σ: output activation function (usually linear or identity).

4.1.4. LayerNorm Standardization

Swin Transformer v2 uses Layer Normalization to stabilize gradients during training.

$$LayerNorm(x) = \frac{x - \mu}{\sqrt{a^2 + \epsilon}} \gamma + \beta$$

Equation 4. LayerNorm Normalization

Where:

- x: input from the previous layer.
- μ are the mean and standard deviation of the features along the dimension of the channels
- $\gamma + \beta$: Learned displacements.
- ϵ: small value to avoid division by zero.

4.1.4. Merging Layers for Resolution Change

To allow for changes in image resolution (usually to get feature maps at lower resolution), layer merging with grouping is applied.

$$x_{output} = Downsample(x) = AveragePool(x, kernel\_size = 2, stride = 2)$$

Equation 5. Merging layers for resolution change

Where:

- x_{output}: Input from the previous block.
- AveragePool: Groups pixels into blocks (for example, 2x2) and calculates the average.
- kernel_size and Stride - determine the size and offset of each grouping.

4.1.5. Loss Function

The Swin Transformer is trained using a stall function that minimizes sorting or prediction error. A cross-entropy loss function is usually used for classification.

$$Loss = -\sum_{i=1}^{N} yi \log(yi)$$

Equation 6. Loss function

Where:

- yi: Actual value for Class I.
- $i = 1$: predicted value for class **I**.
- N: Total number of classes.

4.2. Phase 1: Construction of the Dataset

4.2.1 Definition of Classes

Taking into account the various situations that were mentioned above in the methodology, 10 classes were established that will serve as a guide for the organization and labeling of the images, which are:

- bacterial_leaf_blight
- brown_spot
- healthy
- leaf_blast
- leaf_scald
- narrow_brown_spot
- neck_blast
- rice_hispa
- sheath_blight
- Tungro

4.2.2 Image Collection

Once the classes that will comprise the dataset have been defined, the image collection is carried out. In total, 15,560 images were obtained, distributed equally among the different classes.

4.2.3 Image Conversion

Once the images have been collected, they will be converted to the .jpg format using the Irfanview program.

4.2.4 Balanceo del Dataset

The dataset will then be reviewed to ensure that there is an equitable distribution of images across classes. In case of detecting an imbalance, a technique known as downsampling will be applied to equalize the number of samples in all classes.

To determine how many samples should be removed from a particular class, the difference between the current number of images in that class and the minimum desired number of images per class, generated as a maximum threshold of 1,322 images per class, will be calculated.

The dataset consists of 10 classes. To balance, the class with the fewest images is identified, which in this case is neck_blast with 1322 images. This number will be the maximum threshold of images that the other classes will have. To achieve balance, images of the other classes will be randomly deleted until they equal this threshold.

4.2.5 Image Segmentation

After the balancing process, the Otsu threshold segmentation stage is started then using OpenCV to pre-process the images to detect the spots on the leaves.

The link to the built dataset is attached: https://drive.google.com/drive/folders/1GB8j59CuCg9faKSWBPGcBXNdrZW89kl2?usp=sharing

4.3 Phase 2: Selection of preprocessing methods

4.3.1 Data standardization

Subsequently, a function in python is used that allows the images to be resized to a resolution of 224x224 pixels. For this, bilinear interpolation is used, which is based on the python PIL library.

4.3.2. Data Augmentation

After completing the standardization subphase, rotations will be made in 8 different directions: clockwise (45°, 90° and 135°), counterclockwise (45°, 90° and 135°) and inverted (up and down). For this, the Irfanview program is used again, where the "affine transformation" is applied.

This technique applies a transform matrix to each point of the original image to obtain its new position in the rotated image. The transformation matrix for 90° rotations is as follows: The transformation matrix for an affine rotation in the 2D plane around the origin (0, 0).

By applying these rotations, the initial Dataset of 13,220 images will be expanded to contain a total of 26,440 images. This significant increase in the amount of data will contribute to improving the robustness and generalizability of the data.

4.4 Phase 3: Defining the Deep Learning Architecture

4.4.1. Identification of Architectures

Next, and taking into account the state of the art of research, it is found that architectures such as Swin Transformer (v1 and v2), Vision Transformer (ViT) and Data-efficient Image Transformers (DeiT)

4.4.2 Selection of Architectures

Once the identification stage was completed, the architectures were compared using as the main criterion the precision metric and the values generated in their respective investigations. These precision values will be represented in the following table:

Table II. Accuracy of pre-selected Deep Learning Architectures.

Architecture Name	Dataset Size	Accuracy (%)
Vision Transformer (ViT)	1.2 M images	98.1
Swin Transformer (v1)	1.2 M images	94.9
Swin Transformer (v2)	10 M images	99.1
Data-efficient Image Transformer (DeiT)	1.2 M images	93.2

Taking into account these values, it can be seen that Vision Transformer (ViT), Swin Transformer (v1), Swin Transformer (v2) and Data-efficient Image Transformer (DeiT) are the architectures that present a better percentage of accuracy. Therefore, these will be the ones that will be used in the next phases of the project.

4.5 Phase 4: Training the Model

4.5.1 Dataset Partitioning

After having chosen the deep learning architectures with which to work, the Dataset will be divided into training, validation and testing datasets. Therefore, the number of images in each set would be as follows:

Table III. Distribution of the previously created Dataset.

Dataset Name	Number of Images
Training (train)	18,648
Validation (val)	5,328
Tests	2,664
Total	26,640

4.5.2 Training Setup

Next, the model's training hyperparameters will be established, such as the batch size, which will be 16, 32 and 64, and the number of epochs, which will be 10.

4.5.3 Starting Training

Once the hyperparameters have been configured, training will begin for each of the selected architectures, using the Dataset prepared in the previous phases.

4.5.4 Training Results

Subsequently, taking into account the results obtained from the training of each of the models, they will be represented in the following table and dot graphs:

Table IV. Training Results.

Arq.	**Imgsz**	**Epochs**	**Batch Size**	**Prec. (%)**
Swin Transformer v2	224x224	10	64	99.82
ViT	224x224	10	64	99.35
DeiT	224x224	10	64	99.36

All architectures achieved high performance, with accuracies of over 96% in validation. However, it is observed that the optimized versions of Swin Transformer (v2) and DeiT are suitable for maintaining a balance between training and generalization, with accuracies close to 98% in the validation set. On the other hand, the ViT model showed great potential in accuracy during training, but its performance in validation suggests that it may be prone to overfitting. The training of the models can be found at the following link https://colab.research.google.com/drive/1YBbsvtKK_SMpfRsokwYGFw8Rw6XDwOWn?usp=sharing

The graph titled "Loss Across Epochs" visualizes the loss of training and validation for three different models (Swin Transformer v2, ViT, and DeiT) over 10 epochs. The performance of each model is tracked separately for the training and validation phases. The y-axis represents the loss values, while the x-axis indicates the epochs.

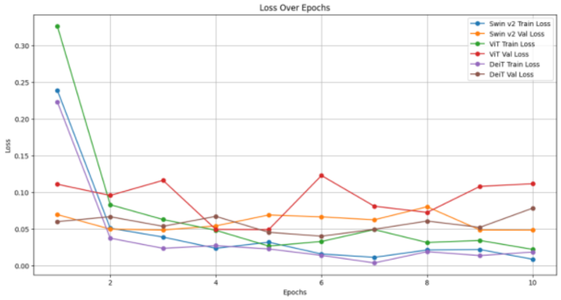


Figure I. Loss over Epochs in the training of three models.

This graph shows the evolution of loss during training and validation over the ages. A rapid decrease in training loss is observed in the early periods, stabilizing with low values from season 5. The validation loss, while not as pronounced, follows a similar pattern and remains stable after some fluctuations. This indicates that the model is learning effectively and generalizing well, with little sign of overfitting.

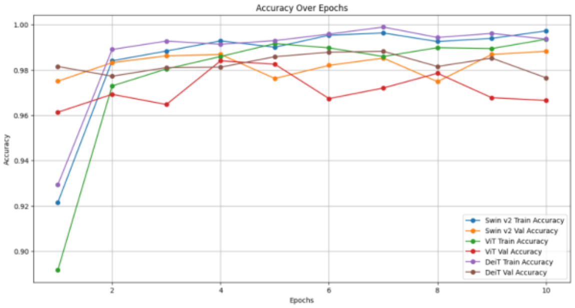
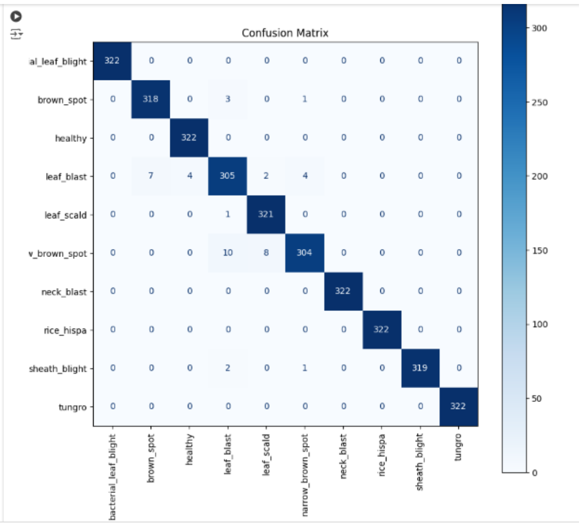


Figure II. Accuracy over epochs in the training of 3 models.

The change in accuracy for the training and validation sets is visualized at each time. Training accuracy increases rapidly and stays close to 100% towards the end, indicating that the model is capturing patterns well. The validation accuracy, although slightly lower, remains at high and stable levels, around 98%, suggesting that the model also performs well on unseen data.

The confounding matrix presents the classification results for each of the 10 classes of diseases and health conditions in rice plants



	bacterial_leaf_blight	brown_spot	healthy	leaf_blast	leaf_scald	narrow_brown_spot	neck_blast	rice_hispa	sheath_blight	tungro
al_leaf_blight	322	0	0	0	0	0	0	0	0	0
brown_spot	0	318	0	3	0	1	0	0	0	0
healthy	0	0	322	0	0	0	0	0	0	0
leaf_blast	0	7	4	305	2	4	0	0	0	0
leaf_scald	0	0	0	1	321	0	0	0	0	0
v_brown_spot	0	0	0	10	8	304	0	0	0	0
neck_blast	0	0	0	0	0	0	322	0	0	0
rice_hispa	0	0	0	0	0	0	0	322	0	0
sheath_blight	0	0	0	2	0	1	0	0	319	0
tungro	0	0	0	0	0	0	0	0	0	322

Figure III. Confusion Matrix for the Classification of Diseases in Rice.

The main diagonal shows the number of correct classifications for each class, indicating that the model has a high level of accuracy for most classes, with values close to 322 correctly classified examples in each.

- The class leaf_blast shows some confusion, with examples misclassified in other classes as brown_spot and v_brown_spot, indicating that these symptoms may be visually similar.
- The v_brown_spot class also presents some confusion, mainly with leaf_blast, which suggests a possible improvement by adjusting the characteristics of these classes or increasing training in these categories.
- The other classes, such as bacterial_leaf_blight, healthy, neck_blast, rice_hispa, and tungro, have a highly accurate classification, with no significant errors of confusion.

4.6 Phase 5: Test Execution

4.6.1 Starting Tests

Based on the results of the training, the models with the best accuracy from each of the previously chosen Deep Learning architectures will be selected. Among them are the model generated by Swin Transformer (v2), with an accuracy of 98.10% using a batch size of 100, Swin Transformer (v1), with an accuracy of 99.28% using a batch size of 32, and Vision Transformer (ViT), with an accuracy of 98.84% using a batch size of 32. These models will be tested using the test dataset, which makes up 10% of the images in the Dataset.

4.6.2 Test Results

After performing the tests, the results obtained will be represented in the following table:

Table V. Test Results.

Architecture Name	Testing Dataset Size	Average Accuracy (%)	Response Time (ms)
Swin Transformer (v2)	1000	99,9	1,8
ViT		99,6	1,2
Data-efficient Image Transformer (DeiT		99,4	2,15

From the tests, three examples were also taken where the operation of each of the models is evidenced.

4.7 Phase 6: Model Evaluation

4.7.1 Discussion

Among the architectures evaluated in previous studies [7], RegNet achieved an accuracy of 90.8%, this performance is attributed to the use of transfer learning. However, in the tests of the current study it is observed that Swin Transformer v2 achieved an accuracy of 99.82% with a response time of 1.8 ms. Similarly, Data-efficient Image Transformer (DeiT) achieved an accuracy of 99.36%, but with a slightly higher response time of 2.15 ms. These results indicate that Swin Transformer v2 and DeiT are highly competitive in terms of accuracy, albeit with a compromise on time efficiency. In contrast, with ViT showing limitations in accuracy, suggesting that recent models have been optimized.

5. Conclusions

The development of the Swin Transformer v2 model for the early detection of brown spot disease in rice plants proved to be effective and robust. The creation of a diverse dataset, which included images of affected and healthy rice under different growing conditions, provided a solid basis for training the model. The selection of advanced preprocessing methods and data augmentation techniques optimized the quality and representativeness of the dataset, increasing the variability of the images and thus improving the generalizability of the model.

The specific architecture of Swin Transformer v2 was carefully tuned to maximize performance in terms of accuracy, recovery and F1 score. This design allowed the model to achieve superior performance in identifying brown spot compared to traditional architectures. The testing tests, carried out under controlled conditions, confirmed that the model offers high accuracy and reliability in the detection of the disease, adapting well to validation and test data.

6. Acknowledgements

Gratitude to Dr. Ferley Medina Rojas, professor at the Universidad Surcolombiana, for his invaluable guidance and advice during the development of this work. Their experience, commitment and constant support were fundamental to the achievement of the objectives of this research.

References

[1] S. Sen, R. Chakraborty, y P. Kalita, "Rice - not just a staple food: A comprehensive review on its phytochemicals and therapeutic potential," *Trends in Food Science & Technology*, vol. 97, pp. 265–285, 2020. doi: 10.1016/j.tifs.2020.01.022.

[2] FAO, "FAOSTAT," *Www.fao.org*, 2021. [Online]. Available at: https://www.fao.org/faostat/es/#data/FBS. [Accessed: Aug 21, 2024].

[3] D. Argelia and R. Córdoba Cantero, "Resilience Capacity of Small Rice Producers in Colombia and Its Implications for Food Sovereignty in the Pandemic Context," *Networks*, vol. 27, no. 1982-6745, 2022. doi:10.17058/redes.v27i1.17946.

[4] A. Cuevas Medina, O. L. Higuera Acosta, and Fedearroz, "Guide for Disease Monitoring and Management," AMTEC, Mass Adoption of Technology, 2018. [Online]. Available in: https://fedearroz.s3.amazonaws.com/media/documents/cartilla_enfermedades_DqWlBTF.pdf.

[5] P. Kartikeyan y G. Shrivastava, "Review on emerging trends in detection of plant diseases using image processing with machine learning," *International Journal of Computer Applications*, vol. 174, no. 11, pp. 39-45, 2021. [Online]. Available in: https://www.researchgate.net/publication/348541626_Review_on_Emerging_Trends_in_Detection_of_Plant_Diseases_using_Image_Processing_with_Machine_Learning.

[6] IASC, "Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images," *IASC*, 2022. [Online]. Available in: https://cdn.techscience.cn/ueditor/files/iasc/TSP_IASC-31-2/TSP_IASC_20679/TSP_IASC_20679.pdf.

[7] T. Lin, Y. Wang, X. Liu, y X. Qiu, "A survey of transformers," AI Open, vol. 3, pp. 111-132, 2022. 10.1016/j.aiopen.2022.10.001. [En línea]. https://www.sciencedirect.com/science/article/pii/S2666651022000146.

[8] T. T. T. Thuy, M. Lübeck, V. Smedegaard-Petersen, E. de Neergaard, and H. J. L. Jørgensen, "Infection Biology of *Bipolaris oryzae* in Rice and Defence Responses in Compatible and Less Compatible Interactions," *Agronomy*, vol. 13, no. 1, p. 231, 2023. doi: 10.3390/agronomy13010231.