AN EXPLAINABLE DEEP LEARNING MODEL FOR CROP DISEASE DECTION

Tran Le Chi Hai[1], Cao Duc Trung[1], Vo Hoang Quan[1], Nguyen Tien Huy[2*]
[1]Ho Chi Minh City University of Education
[2]VNUHCM - Ho Chi Minh City University of Science

ARTICLE INFO

Received:	**18/10/2024**
Revised:	**22/11/2024**
Published:	**22/11/2024**

KEYWORDS

Explainable AI (XAI)
Deep Learning
Transfer Learning
Leaf Disease
Convolutional Block Attention Module

ABSTRACT

In recent years, many studies have applied deep learning in artificial intelligence to assist in the detection and classification of plant diseases. However, these models, when applied in practice, often lack transparency and suffer from insufficient accuracy. In this paper, we use two explainable artificial intelligence (XAI) techniques to analyze how the model identifies diseases, providing explanations for predictions using the New Bangladesh Crop dataset, which is derived from the Plant Village dataset and focuses on key food crops. To evaluate the model's focus on diseased regions, we calculate the Intersection over Union (IoU) values for selected disease images from each crop. The experimental results guide the selection of appropriate XAI methods and help fine-tune the model for improved accuracy. We propose an enhanced VGG16 model with attention mechanisms, achieving relatively high accuracy and improved focus on diseased regions of plant leaves.

MÔ HÌNH HỌC SÂU GIẢI THÍCH ĐƯỢC CHO BÀI TOÁN PHÂN LOẠI BỆNH CÂY TRỒNG

Trần Lê Chí Hải[1], Cao Đức Trung[1], Võ Hoàng Quân[1], Nguyễn Tiến Huy[2*]
[1]Trường Đại học Sư phạm Thành phố Hồ Chí Minh
[2]Trường Đại học Khoa học Tự nhiên - Đại học Quốc gia Thành phố Hồ Chí Minh

THÔNG TIN BÀI BÁO

Ngày nhận bài:	**18/10/2024**
Ngày hoàn thiện:	**22/11/2024**
Ngày đăng:	**22/11/2024**

TỪ KHÓA

Trí tuệ nhân tạo có thể giải thích
Học sâu
Học chuyển giao
Bệnh trên lá
Mô-đun chú ý khối tích chập

TÓM TẮT

Trong những năm gần đây, nhiều nghiên cứu đã áp dụng học sâu trong trí tuệ nhân tạo nhằm hỗ trợ nhận diện và phân loại bệnh cây trồng. Tuy nhiên, các mô hình này khi áp dụng thực tế thường thiếu minh bạch và có độ chính xác chưa cao. Trong bài báo này, chúng tôi sử dụng hai kỹ thuật trí tuệ nhân tạo có thể giải thích (XAI) để phân tích cách mô hình nhận diện bệnh, qua đó cung cấp giải thích cho các dự đoán dựa trên bộ dữ liệu bệnh cây trồng New Bangladesh Crop được lấy ra từ bộ dữ liệu Plant Village ở một số loại cây lương thực trọng điểm. Để đánh giá khả năng tập trung của mô hình CNN vào vùng bệnh, chúng tôi tính toán giá trị Intersection over Union (IoU) cho một số ảnh bệnh của từng loại cây. Kết quả thực nghiệm giúp định hướng việc lựa chọn phương pháp XAI phù hợp và tinh chỉnh mô hình nhằm tăng độ chính xác. Chúng tôi đề xuất mô hình VGG16 cải tiến với cơ chế chú ý, đạt độ chính xác tương đối cao và khả năng tập trung vào vùng bệnh trên lá cây được cải thiện.

DOI: https://doi.org/10.34238/tnu-jst.11353

[*] Corresponding author. *Email: ntienhuy@fit.hcmus.edu.vn*

1. Introduction

The role of Artificial Intelligence (AI) in agriculture has become increasingly prominent, particularly in the detection and classification of crop diseases, which are critical for improving productivity and ensuring food security. Leaves are essential indicators in identifying plant diseases, as symptoms or damage often appear on them. The application of AI for leaf-based disease classification is crucial, along with the need for model interpretability to enhance trustworthiness. Convolutional Neural Networks (CNNs) have achieved significant success and are being extensively applied in the classification of food crop diseases based on leaves. However, these models face limitations, such as the requirement for large, high-quality datasets and their "black-box" nature, which makes it difficult for users to understand how decisions are made. Therefore, the development and application of explainable AI (XAI) methods are essential for practical deployment.

In practice, plant disease classification is a challenging task. Hughes and Salathe introduced the PlantVillage dataset, which consists of over 50,000 color images of varying sizes collected from both healthy and diseased crops [1]. In the study by Mohanty et al. [2], widely used CNN models such as AlexNet [3] and GoogLeNet [4] were applied to predict plant diseases based on images from the PlantVillage dataset, achieving high accuracy. However, when tested on images from reliable online sources, the accuracy dropped to just over 31%. Russel and Selvaraj [5] conducted research using both the PlantVillage dataset and the Mepco TrophicLeaf dataset. They employed multiple CNN-based deep learning models to create new image feature filters, such as Law’s Mask, which can self-adjust to the disease stages of leaves, enhancing disease recognition. Ferentinos [6] presented the implementation of pre-trained models for the classification of 25 different plant species and 58 different diseases. Using the extended PlantVillage dataset along with additional images, the author achieved an accuracy of 99.53% with a pre-trained VGG model, showing potential for real-world system scalability. Transfer learning methods were used in the research by Zhang et al. [7] to classify nine types of maize leaves. By utilizing max-max-ave pooling in three hidden layers of the CNN, the authors achieved accuracy rates of 98.9% for GoogleLeNet and 98.8% for the Cifar10 neural network. In another study [8], the author employed various models combined with XAI methods to verify results and analyze the strengths and limitations using a tomato crop dataset. Marco de Benito Fernández and colleagues, in their paper [9], detailed how CNN models function and evaluated post-hoc XAI methods, specifically LIME, SHAP, and Grad-CAM, to assess CNN model results. Machine learning and CNN models were compared and tested for real-world applicability as discussed in [10].

Based on the above prominent studies, there are still methodological and knowledge gaps that need to be addressed. Previous research has largely focused on maximizing CNN performance for crop disease identification and classification using various techniques but has not sufficiently explored algorithms that enhance model interpretability. Some other papers have concentrated on XAI methods to explain model results based on pre-trained models. However, the authors did not focus extensively on improving models to suit datasets specific to particular tasks. In general, numerous studies have been conducted to optimize and support plant disease classification, with each study focused either on large datasets or on a specific crop type. In this research, we propose a VGG16 model, which has been widely used in previous studies, combined with the CBAM module for key food crops across multiple countries. This approach aims to increase diversity beyond a single crop. We then evaluate the model using two popular XAI methods, Grad-CAM and XRAI, and assess model attention using Intersection over Union (IoU) to compare the selected XAI methods. Figure 1 shows the architecture of the proposed model. The outcome of the study is the proposed VGG16CBAM model architecture and evaluation metrics using two XAI algorithms, laying the groundwork for selecting suitable post-hoc XAI algorithms.

In the following sections, we provide details of the dataset, research methodology, and experimental results.

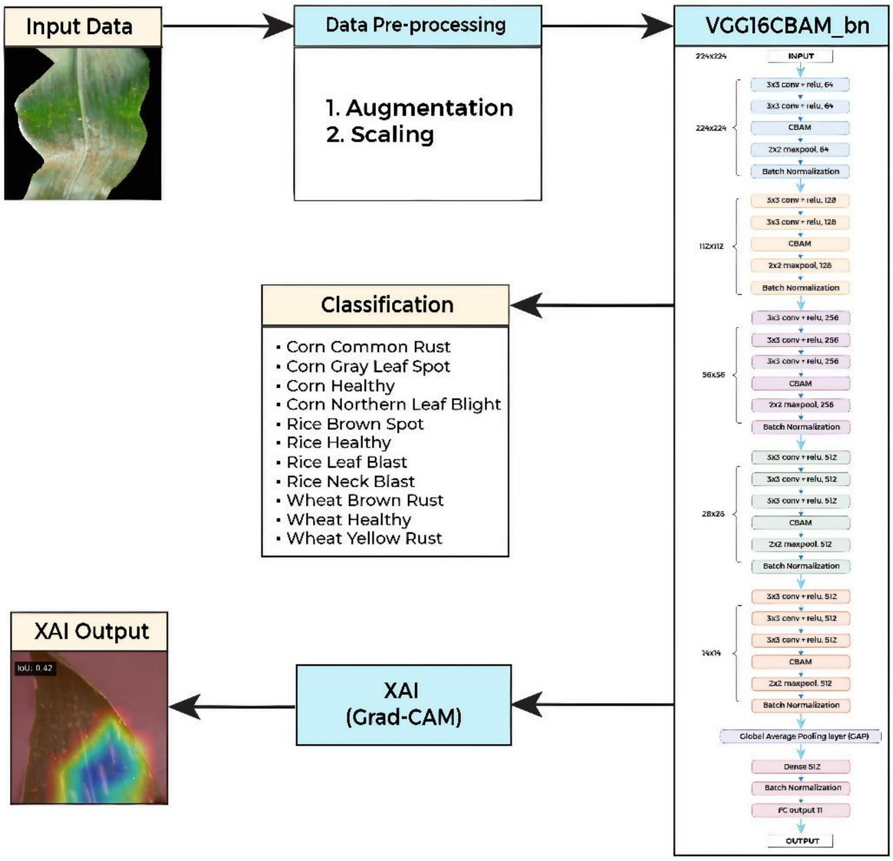

Figure 1. *Proposed system model framework*

2. Methods

2.1. VGGNet

The VGGNet model was developed by the Visual Geometry Group at the University of Oxford. It is a groundbreaking CNN that paved the way for subsequent achievements in the field of computer vision. VGGNet utilizes 3x3 convolutional layers and 2x2 pooling layers, resulting in a simple yet effective architecture. This architecture allows for increasing the depth of the network for various tasks to improve accuracy. Additionally, VGGNet employs three fully connected layers with 4096 channels (first two layers) and 1000 channels (last layer) for image classification. ReLU is used as the activation function for the hidden layers. The two most popular versions of VGGNet are VGG16 and VGG19, which achieve comparable accuracy on various datasets. In this study, VGG16 was chosen as the basis for the CNN network combined with CBAM to optimize image recognition performance.

2.2. Convolutional Block Attention Module

In the CBAM architecture, Sanghyun Woo et al. [11] proposed an integrated attention mechanism to enhance the local and global feature representation capability of convolutional neural network (CNN) models. CBAM utilizes a channel attention module and a spatial attention module to focus on the important features in an image. This enables the model to concentrate on critical information and effectively capture complex dependencies and long-range relationships in image data. The complete CBAM module is illustrated in Figure 2.

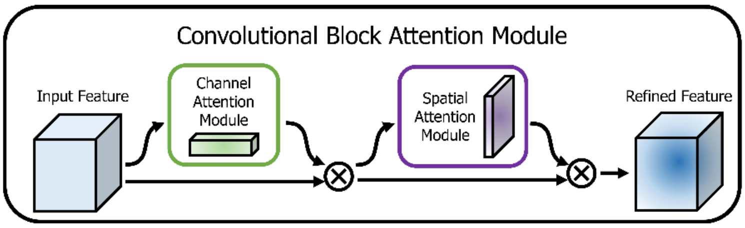

Figure 2. *CBAM module* [11]

Channel Attention Module: The proposed method is based on the correlation between features in data channels. The main objective is to focus on the important factors in the input data. To obtain the $\boldsymbol{F}_{avg}^{c}$ and $\boldsymbol{F}_{max}^{s}$ values, spatial information from the feature maps undergoes average and max pooling operations. These values are then fed into a $\mathbf{M}_c \in \mathbf{R}^{C\times1\times1}$ network with one hidden layer. Channel attention is determined using equations 1 and 2.

$$\mathbf{M}_c(\mathbf{F}) = \sigma\ (\text{MLP}(\text{AvgPool}(\mathbf{F}) + \text{MLP}(\text{MaxPool}(\mathbf{F}))) \quad (1)$$

$$\mathbf{M}_c\ (\mathbf{F}) = \sigma\ (\mathbf{W}_1\ (\mathbf{W}_0\,(\boldsymbol{F}_{avg}^{c})) + \mathbf{W}_1\ (\mathbf{W}_0\,(\boldsymbol{F}_{max}^{c}))) \quad (2)$$

Here, σ denotes the sigmoid function, $\mathbf{W}_0$ and $\mathbf{W}_1$ represent the weight values, and MLP stands for multilayer perceptron network.

Spatial Attention Module: This module exploits the relationship between features in the spatial domain to generate a spatial attention map. The spatial module focuses on the locations containing important information in the input. To generate the spatial attention maps, max and average pooling operations are performed, resulting in the corresponding values of $\boldsymbol{F}_{avg}^{c} \in \mathbf{R}^{1\times H\times W}$ and $\boldsymbol{F}_{max}^{s} \in \mathbf{R}^{1\times H\times W}$. The obtained values are then concatenated and applied to a convolutional layer as $\mathbf{M}_s\,(\mathbf{F}) \in \mathbf{R}^{H\times W}$. Equations 3 and 4 are used to generate the spatial map.

$$\mathbf{M}_s\,(\mathbf{F}) = \sigma\ (f^{7x7}([\text{AvgPool}(\mathbf{F});\ \text{MaxPool}(\mathbf{F})])) \quad (3)$$

$$\mathbf{M}_s\ (\mathbf{F}) = \sigma\ (f^{7x7}([\boldsymbol{F}_{avg}^{s};\ \boldsymbol{F}_{max}^{s}])) \quad (4)$$

Here, f^{7x7} denotes the 7x7 convolution operation. This attention mechanism not only indicates where to focus but also improves the representation of important features. The CBAM module in this paper is used to emphasize meaningful features both in the spatial and channel dimensions, focusing on important features and suppressing unnecessary ones. We used the channel module first and then the spatial module, which yielded the best results for our task.

2.3. Modified VGG16CBAM_bn model

We added a CBAM layer after the last convolutional block, increasing the depth of the original VGG16 network and allowing the model to learn more complex information. The CBAM layer takes the output of the convolutional layer as input, aggregates the important information extracted by the model through each block, and utilizes the knowledge of the model pre-trained on the large ImageNet dataset to refine the feature map to fit the data. The architecture of the proposed VGG16CBAM_bn model is shown in Figure 3. We also added a Batch Normalization layer at the end of each block of the original VGG16 model to avoid overfitting and mitigate vanishing and exploding gradients. The BatchNorm layer helps to increase the reliability of the model by normalizing the input of each block in the network, and also speeds up the training process of the model. Adding a BatchNorm layer after each convolutional block has been shown to be effective when applied to the VGG model [12], [13]. We replaced the Fully Connected layer with a GAP layer by averaging the final feature maps spatially, which reduces the number of parameters and the size of the trained model.

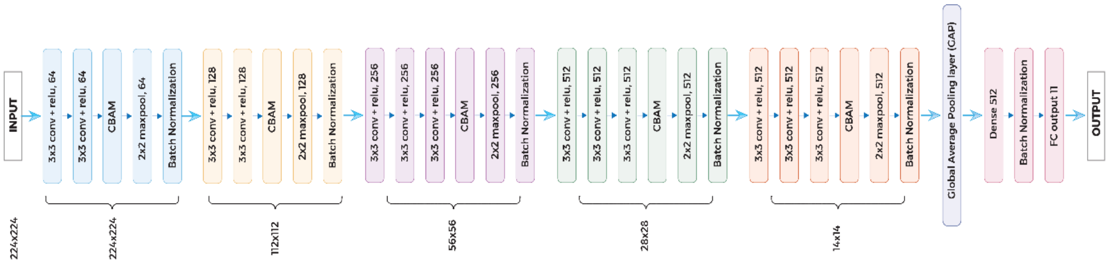

Figure 3. *VGG16CBAM_bn model*

We compared the proposed method with several other models to evaluate the accuracy of the model using the introduced dataset of key food crop diseases. In this study, we used the transfer learning approach, where the convolutional layers of the proposed model utilize pre-trained knowledge with the ImageNet dataset to accelerate the training process and improve accuracy for the task data. Although deep learning can automatically extract features, we still need to know which features are useful and have a significant impact on the model's decision. CNNs have the potential to find out which features are relevant and need to be considered. We conducted experiments with several post-hoc explanation methods for visual models to address this issue.

3. Experiments and Results

3.1. Dataset

We evaluated the proposed model on a dataset comprising three important food crops: maize, rice, and wheat. These images were gathered from various sources, including research papers and Kaggle. Additionally, we manually labeled a subset of images from the diseased plant class to evaluate the IoU metric.

- The dataset includes 11,609 images, categorized into eight diseased and three healthy classes. It contains 4,078 rice images, 3,852 maize images, and 3,679 wheat images. Notably, the maize images from PlantVillage [1] have a black background, while the images of rice and wheat generally feature varied, natural-looking environments.
- The dataset is split into 60% for training, 20% for validation, and 20% for testing, resulting in 6,965 training images, 2,322 validation images, and 2,322 test images. For segmentation tasks, we randomly sampled 10 images from each disease class, resulting in 80 annotated images across the eight disease categories. These annotations were used to evaluate the model's segmentation performance.

3.2. Experimental Setup

Initially, we did not have predefined splits for training, validation, and testing. Thus, the data was partitioned using a seed value of 123 to ensure consistency across experiments.

- For the CNN, we utilized pre-trained weights from the VGG16 model and removed the classification layer. We added 1 Batch Normalization layer after each block and changed the number of output classes in the softmax layer to 11.
- For CBAM, we used a ratio of 16 and a kernel size of 3x3 in the Dense layer, responsible for combining the 2 Channel Attention and Spatial Attention modules.

The training and experimental environment was set up on Google Colab with 12.68 GB of RAM and two NVIDIA Tesla T4 GPUs. The training process utilized a gradient descent algorithm with the following hyperparameters: optimizer = Adam, learning rate = 1e-4, momentum = 0.9, batch size = 32, and epochs = 40. The learning rate was reduced by a factor of 0.5 after 3 epochs if the validation loss did not show significant improvement. The total training time across the 40 epochs was 3 hours and 12 minutes. Our proposed model, VGG16CBAM_bn, has a size of 60.13 MB and is saved in .h5 format using the TensorFlow framework. Table 1 presents the average prediction time parameters for two XAI algorithms on a single input image from the testing set.

Table 1. *Prediction time of the two XAI methods (measured in seconds)*

XAI methods	Predict class	Predict heatmap
Grad-CAM	0.13s	0.47s
XRAI	0.14s	23.00s

We compared our model (VGG16CBAM_bn) to others such as ResNet50, EfficientNetV2B0, and MobileNetV2. As shown in Table 2, our model outperformed them in terms of accuracy, highlighting the benefits of adding CBAM and Batch Normalization.

Table 2. *Performance comparison of different models*

Model	Parameters	Accuracy
ResNet50	25.6M	91.27
VGG16	14.72M	90.96
EfficientnetV2B0	7.2M	93.46
MobileNetV2	3.5M	90.23
ResNet50 + CBAM	28.09M	93.57
VGG16CBAM_bn (Our Proposed)	15.07M	94.75

The performance of the proposed model on each class is presented in Table 3, which details the precision, recall, and F1-score for each class, highlighting its effectiveness in distinguishing between healthy and diseased crops.

Table 3. *Model performance on each class*

Class	Precision	Recall	F1-Score
Corn Common Rust	1.00	1.00	1.00
Corn Gray Leaf Spot	0.90	0.93	0.91
Corn Healthy	1.00	1.00	1.00
Corn Northern Leaf Blight	0.97	0.94	0.96
Rice Brown Spot	0.78	0.76	0.77
Rice Healthy	0.86	0.92	0.89
Rice Leaf Blast	0.85	0.76	0.80
Rice Neck Blast	1.00	1.00	1.00
Wheat Brown Rust	0.99	0.99	0.99
Wheat Healthy	0.90	1.00	1.00
Wheat Yellow Rust	1.00	0.99	0.99

3.3. Visualization Results

3.3.1. Heatmaps Comparison

Figure 4 shows images with three input channels used by the model. Grad-CAM and XRAI were applied to the CNN's output layer (block5_conv3) to produce channel-wise heatmaps, highlighting regions important for predictions. Grad-CAM provides global localization but often focuses on only part of the diseased area or misidentifies the central region. XRAI, using pixel-level attribution and a large K value (1200), segments critical regions and filters out irrelevant areas, offering finer-grained localization. While Grad-CAM highlights general regions, XRAI improves accuracy by focusing more precisely on diseased areas.

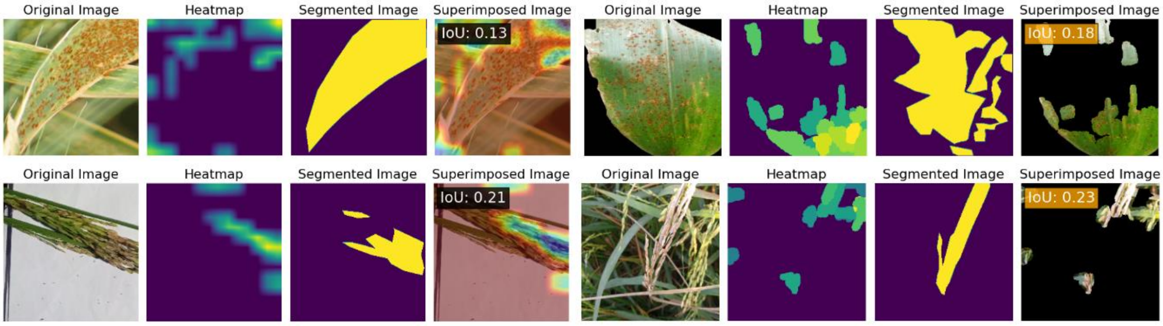

Figure 4. *Visualization images with Grad-CAM and XRAI of the VGG16CBAM_bn architecture (left side is Grad-CAM, right side is XRAI)*

3.3.2. IoU Comparison

We labeled 80 randomly selected images across 8 disease classes of plants to implement the Intersection over Union (IoU) evaluation method for the VGG16CBAM_bn and VGG16-based models. Figure 5 visualizes the threshold values corresponding to each disease class.

The threshold was determined based on the ratio of the pixel count of the annotated disease region to the total pixel count of the corresponding 2D image. This approach enables an objective assessment of the model's focus, as the output of the XAI algorithms—Grad-CAM and XRAI—are 2D heatmaps. By employing the IoU evaluation method, we could directly compare the model-identified important regions to the annotated disease areas.

Grad-CAM and XRAI produce heatmaps with varying color intensities representing importance percentiles ranging from 0 to 1. To ensure meaningful comparison, we selected percentiles corresponding to the most critical regions as identified by the model, aligning them closely with the labeled disease areas. The selected percentiles for each class were derived from their respective threshold ratios. For instance, Rice Leaf Blast and Rice Brown Spot, with thresholds of 2.26% and 3.31%, resulted in an average percentile of 0.97. Similarly, Wheat Brown Rust, with a threshold of 23.21%, was assigned a percentile of 0.77. This method ensures that the selected critical regions in the heatmaps have areas closely matching the annotated regions, facilitating accurate model evaluation.

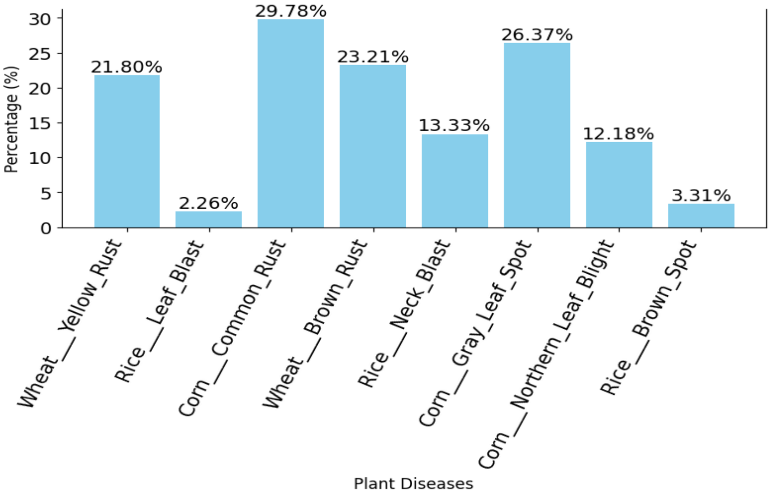

Figure 5. *The average number of pixels for each layer with 80 label samples*

The IoU values are summarized in Table 4. Grad-CAM showed little variance between VGG16 and VGG16CBAM_bn, implying that it relies heavily on interrelated regions rather than distinct disease features. Conversely, XRAI showed higher IoU values in several classes for

VGG16CBAM_bn, demonstrating that the CBAM module enhances the model's ability to focus on relevant features, improving both recognition and performance. High accuracy is positively correlated with higher IoU values, reflecting better focus on important disease regions.

Table 4. *Average IoU with Threshold for Basic VGG16 and Enhanced VGG16 (80 samples with 8 classes, 10 images per class)*

	Grad-CAM		**XRAI**	
Class	**VGG16**	**VGG16CBAM**	**VGG16**	**VGG16CBAM**
Corn Common Rust	0.072	0.068	0.206	0.223
Corn Gray Leaf Spot	0.131	0.217	0.197	0.253
Corn Northern Leaf Blight	0.258	0.150	0.177	0.205
Rice Brown Spot	0.032	0.04	0.143	0.227
Rice Leaf Blast	0.156	0.146	0.21	0.382
Rice Neck Blast	0.241	0.212	0.142	0.289
Wheat Brown Rust	0.139	0.121	0.273	0.124
Wheat Yellow Rust	0.089	0.176	0.073	0.293
Mean	**0.140**	**0.142**	**0.178**	**0.250**

4. Conclusion

In this study, we introduced a novel model improvement with a conventional CNN using the CNN architecture integrated with CBAM. This approach captures both local and global dependencies in the image. Our model outperforms some conventional CNN models for the critical crop disease dataset. Additionally, we collected and curated 11,609 real-world crop disease images, labeled some test samples to evaluate the model's ability to focus on important regions. We also experimented with several XAI methods for computer vision tasks to analyze how the model identifies diseases, laying the groundwork for further developments in CNNs.

As discussed in the error analysis section, we will improve performance in cases of misidentifying important regions along with the ability to focus on multiple disease regions on leaves. Additionally, we are also collecting more data and enhancing the model's interpretability for computer vision tasks. It is expected to provide better explanations for deep learning models.

REFERENCES

[1] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060,* 2015.

[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science,* vol. 7, p. 1419, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems,* vol. 25, pp. 1097-1105, 2012.

[4] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[5] N. S. Russel and A. Selvaraj, "Leaf species and disease classification using multiscale parallel deep CNN architecture," *Neural Computing and Applications,* vol. 34, no. 21, pp. 19217-19237, 2022.

[6] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture,* vol. 145, pp. 311-318, 2018.

[7] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access,* vol. 6, pp. 30370-30377, 2018.

[8] S. Kiriella, S. Fernando, S. Sumathipala, and E. Udayakumara, "Explainable AI techniques for Deep Convolutional Neural Network based plant disease identification," in *8th International Conference on Information Technology Research (ICITR),* 2023, pp. 1-6.

[9] M. D. B. Fernández, D. L. Martínez, A. González-Briones, P. Chamoso, and E. S. Corchado, "Evaluation of XAI Models for Interpretation of Deep Learning Techniques' Results in Automated

Plant Disease Diagnosis," in *Sustainable Smart Cities and Territories International Conference*, Springer, 2023, pp. 417-428.

[10] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *Journal of Big Data,* vol. 11, no. 1, p. 5, 2024.

[11] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3-19.

[12] S. Ioffe, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

[13] S. Liu *et al.*, "Convolutional normalization: Improving deep convolutional network robustness and training," *Advances in Neural Information Processing Systems,* vol. 34, pp. 28919-28928, 2021.