Rice Leaf Diseases Classification using CNN and Swin Transformer with Transfer Learning.

*

Sajidul Islam Ronju
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
Email: 1803161@student.ruet.ac.bd

Prof. Dr. Md. Rabiul Islam
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
Email: rabiul_cse@yahoo.com

Faysal Ahamed
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
Email: faysalahamed@ece.ruet.ac.bd

***Abstract*—Rice is one of the major cultivated crops in Bangladesh which is affected by various diseases at various stages of it's cultivation. However, it is challenging for farmers to accurately identify these diseases due to their limited knowledge. Recent advancements in deep learning have demonstrated the potential of Convolutional Neural Networks (CNN) for automatic image recognition, which can greatly benefit in addressing such issues. As the availability of rice leaf disease image dataset is limited, a medium-sized dataset have been collected from internet which contains 18445 images from the internet and experiment was conducted on ten major diseases namely bacterial_leaf_blight(0), brown_spot(1), healthy_leaves(2), leaf_blast(3), leaf_scald(4), narrow_brown_spot(5), neck_blast(6), rice_hispa(7), sheath_blight(8), tungro(9). In this study, a pre-trained and fine-tuned deep learning model was employed. After conducting experiments with a total of 18,445 images, it was noted that the fine-tuned CNN and Swin Transformer architecture exhibited superior performance. The fine-tuned model achieved a peak training accuracy of 97.21% and testing accuracy of 96.40%, while the pre-trained VGG-16 model yielded 92.42% training accuracy and 91.83% testing accuracy. Furthermore, the CNN and Swin Transformer models demonstrated lower training and validation loss compared to the pre-trained VGG-16, with values of 0.1660 and 0.1970 respectively. Despite this, the CNN and Swin Transformer models utilized a larger number of parameters at 1955M, whereas VGG-16 only used 211M. It is important to note that this enhanced performance of the CNN and Swin Transformer models is accompanied by increased computational requirements, as the Swin Transformer is a more complex and parameter-intensive model compared to the pre-trained VGG-16 with transfer learning. Lastly, it was observed that the CNN and Swin Transformer models perform better when the dataset size is larger.**

***Index Terms*—Convolutional Neural Network, Fine Tuning, Transformer, Deep Learning, Rice Leaf Diseases, Transfer Learning.**

I. Introduction

Rice is the staple source of food in Bangladesh as well as across the world. It is attacked by a variety of diseases in various stages of its cultivation. Therefore, early detection and remedy of such diseases are beneficial to ensure high quantity and best quality but this is very difficult due to the huge expanse of land under individual farmers and the huge variety of diseases as well as the occurrence of more than one disease in the same plant. Agricultural expert knowledge is not accessible in remote areas and it is a time taking process [1]. It is important that disease symptoms are identified early and accurately for undertaking disease control actions or remedial measures. The symptoms of rice diseases, such as rice bacterial leaf blight, rice sheath blight and rice blast, appear initially as spots around the infected areas. So the detection of these diseases mainly relies on their spots. At present, farmers or technicians mainly rely on their experiences, the guide books and experts to identify these diseases. It is suitable for detecting some familiar and typical diseases. However, some different diseases can cause similar spots and the same diseases can cause different spots because of different rice varieties and local conditions. It increases the complexity of identifying rice diseases [2]. So with the high amount of production and consumption, it is also very much prone to the various types of plant diseases which will have negative consequences on the quality and quantity of rice production [3]. Therefore, we have come up with an automated system, which will contribute to the development of Bangladesh's agriculture sector. Our proposed system will automatically detect if a leaf is healthy or infected from the leaf's image [4]. Out of the aforementioned papers, we have selected the research conducted by Shreya et. al (2020) in order to enhance their research outcomes. The utilization of performance metrics was not comprehensive, and the accuracy of their proposed VGG-16 model based on CNN was relatively low at 92%. Additionally, they did not mention the computational parameters and hyperparameters employed. Furthermore, their model was trained on a small

dataset consisting of only 1649 images of rice leaf diseases. In order to overcome these limitations, we have devised and implemented our proposed methodology and dataset to address this issue.
Followings are the contributions of our research:

- Acquired an extensive dataset containing images of rice leaf diseases.
- Implemented a CNN model utilizing swin transformer architecture to enhance classification accuracy and minimize training and validation loss.

Section-II covers some works related to rice leaf disease detection and classification with their contributions and limitations. In Section-III, The dataset and suggested methods are covered. It also provides a thorough explanation of the suggested architecture, model training, and data pre-processing. Section-IV presents the details description of the dataset. Section-V compares the suggested architecture with related works and examines the experimental results and performance of the design. This research has come to an end with Section-VI. The results of our research are summarized and the future work in this chapter. We have also attempted to draw attention to our work's shortcomings and possible areas for improvement in the future.

II. RELATED WORK

The CNN architecture VGG16 has been used for training and testing on datasets gathered from the internet as well as paddy fields. Applying this model resulted in 92.42% accuracy. The three main diseases—bacterial leaf blight, brown spot, and rice leaf blast—were the focus of the investigation, which examined 1649 images of rice leaves [1]. TThis work developed the SVM approach, which successfully identified data based on form, color, and texture aspects using 108 training examples. With an accuracy of 97.2%, the results showed that SVM was successful in identifying and classifying illness areas [2].
The authors' work employed CNNs approaches to analyze eight important rice diseases, including bacterial leaf blight, false smut, rice hispa, blast, stemborer, sheath blight, brown spot, and brown planthopper. The report is divided into two main sections: the first part describes the research's survey methodology, and the second part explores the most advanced method for detecting rice disease (RDD) using CNNs. [3].
In order to classify rice plant illnesses based on photographs, the study evaluated a variety of pre-trained deep CNN models, such as AlexNet, Vgg16, ResNet152V2, InceptionV3, InceptionResNetV2, Xception, MobileNet, DenseNet169, NasNetMobile, and NasNetLarge. The research used a dataset consisting of 1216 photos of damaged rice plants from actual fields; the photos were divided into seven classes: brown spot, rice blast, bacterial leaf blight, sheath rot, fake smut, and healthy leaves. Of all the models that were evaluated, Vgg16 had the best classification accuracy, coming in at 93.11% [4].
The authors' work employed CNNs approaches to analyze eight important rice diseases, including bacterial leaf blight, false smut, rice hispa, blast, stemborer, sheath blight, brown spot, and brown planthopper. The report is divided into two main sections: the first part describes the research's survey methodology, and the second part explores the most advanced method for detecting rice disease (RDD) using CNNs [6].

III. PROPOSED METHODOLOGY

The technique described in Shreya et al. (2020) [1] has been applied to the proposed framework, with the inclusion of stages as shown in Figure 3. To improve data efficacy and consistency, a larger dataset was added, and picture normalization was used. We used the suggested dataset to create our proposed CNN with Swin Transformer architecture. We evaluated our model's performance by adjusting its hyperparameters. Figure 3 presents an excellent illustration of the complete process that demonstrates the seamless succession of implementation phases.

Significant variations in superstructures and equipment were revealed by a study of rice leaf disease photos, underscoring their unique characteristics. A CNN-Swin Transformer model was created to combine the advantages of both architectures for rice leaf disease image analysis since CNNs are good at extracting local features and transformers with self-attention are good at capturing particular characteristics.Figure 3 displays the structure of the model network. The three parts of the model are the CNN backbone, transformer backbone, and multi-layer perceptron (MLP) neural network-based classification module. In order to reliably and effectively extract image features, the model makes use of the CNN structure for feature extraction and a CNN backbone.

The CNN backbone generates a feature vector by continually stacking CNN-Blocks, which enriches the lowest layer of the feature map with more semantic information. After employing a CNN-Block in the CNN backbone to reduce the resolution of the feature map, four layers are used to extract data. Each layer has 2, 2, 6, 2 CNN-Blocks; the first CNN-Block in each layer employs a convolutional kernel with a two-pixel stride in order to lower the resolution. This method was created using the research [5] as a basis.It appears from empirical data that replacing pooling layers with convolutional layers that have strides longer than one pixel does not reduce classification accuracy—in fact, it may even somewhat increase it. As a result, the feature map size is condensed using a stride of two pixels, guaranteeing improved accuracy, a smaller spatial dimension, and the removal of duplication. The transformer backbone is supported by the Swin transformer. Initially, each visual channel is split up into 48 patches, each measuring four by four. These patches are called "tokens" because they are the smallest processing units in the model, making up the full image without any overlap. The transformer backbone then uses encoders to obtain picture attributes after embedding each patch. Local information is included in the feature vector that is produced. Moreover, the transformer backbone extracts features via patch merging before to each encoder, producing feature maps that include high-level semantic information.

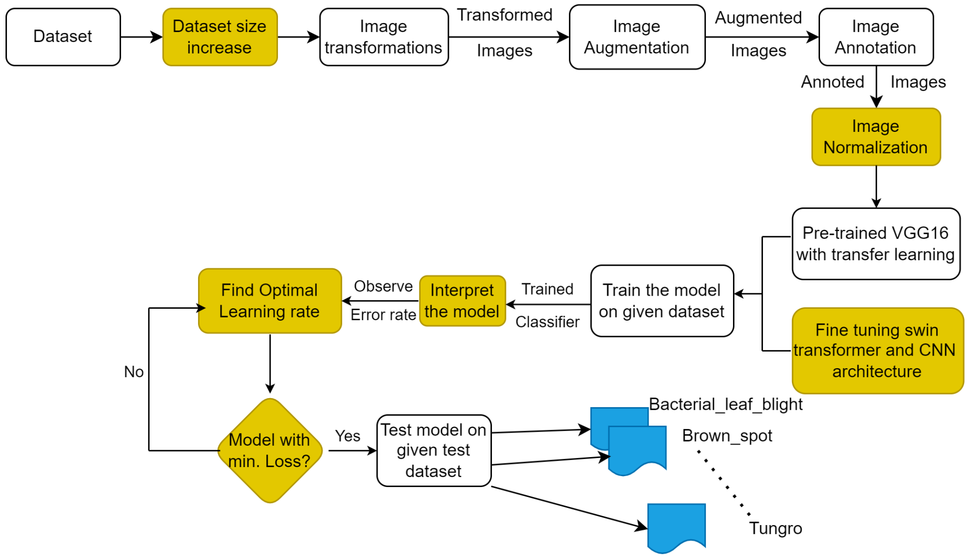


Fig. 1: Proposed CNN based architecture.

During each patch merging process, tokens undergo a downsampling by a factor of two in order to concatenate the properties of adjacent 2×2 tokens [7].

Three fully connected layers make up the MLP neural network: an output layer whose size is dependent on the number of classes to be detected, a hidden layer with 128 dimensions, and an input layer with 1024 dimensions. The CNN-Swin model maps the fused feature vectors to rice classification using an MLP neural network [8].

The Backbone of a Convolutional Neural Network (CNN) In order to minimize overfitting and provide better results, the CNN backbone of the model should be a condensed CNN structure like the VGG model [9], which has minimal parameters. On ImageNet, however, VGG-19's Top-1 accuracy is 72%, and on the military ship dataset utilized for this investigation, it was 84.5%. Both have less classification accuracy when compared to the sophisticated CNN models of today. The CNN backbone utilized in this study created Resnet-based models by taking cues from Resnet [10]. Better results are obtained with the multi-branch CNN-Block module, which is based on CNN structures with fewer parameters (like the VGG model).

Cascaded CNN-Blocks are used by the CNN backbone to extract features from the picture. It is split into two primary sections. The initial segment comprises distinct CNN-Blocks designed to reduce picture resolution, a 3×3 convolutional layer, a BN (batch normalization) layer, a parallel structure made up of a BN layer and a 1×1 convolutional layer, and so on. To create the feature map, a rectified linear (ReLU) activation function is then employed. The final result is the output of the second portion, which consists of a global pooling layer and four layers with different numbers of Resnet blocks in each layer. In Figure 4.5, the Layer architecture is displayed. The convolution channel counts of 64, 128, 256, and 512, respectively, are the only differences between Layer 1 and Layers 2, 3, and 4. Furthermore, Layer 3's volume is three times that of Layer 1's. Several CNN-Blocks with comparable structural patterns comprise each layer. The main distinction is shown in the first CNN-Block, where variations in the resolution and channel numbers relative to the input X are caused by the 2 pixel strides in the 3×3 convolution layer and the 1×1 convolutional layer. In the ensuing CNN-Blocks, the 1×1 and 3×3 convolutional layers have one pixel strides, guaranteeing that the feature map's dimensions and channel counts don't change.

We have also developed a CNN model with transfer learning with 4 Convolution layers each of which is followed by ReLU, Maxpooling and dropout layer followed by 2 Fully Connected Layer and SoftMax. As shown in Figure 4, VGG-16 architecture is a deep convolutional network with weights pre-trained on the ImageNet Database which contains 3.2 million clearly annotated images of 5247 categories [11]. Thus knowledge in form of weights, architecture and features learnt on one domain can be transferred to another domain by Transfer Learning on such pre-trained models. The features are generic in the early layers and more dataset-specific in the later layers. In our model, the initial 5 blocks of convolution layers are frozen for behaving as a feature extractor which is the advantage of CNN over traditional techniques and the last dense layer of size 128 followed by softmax layer of size 4(since number of classes is 4) is used for classification. The pre-trained VGG-16 model is again trained on our dataset and fine-tuned to get the classifications.

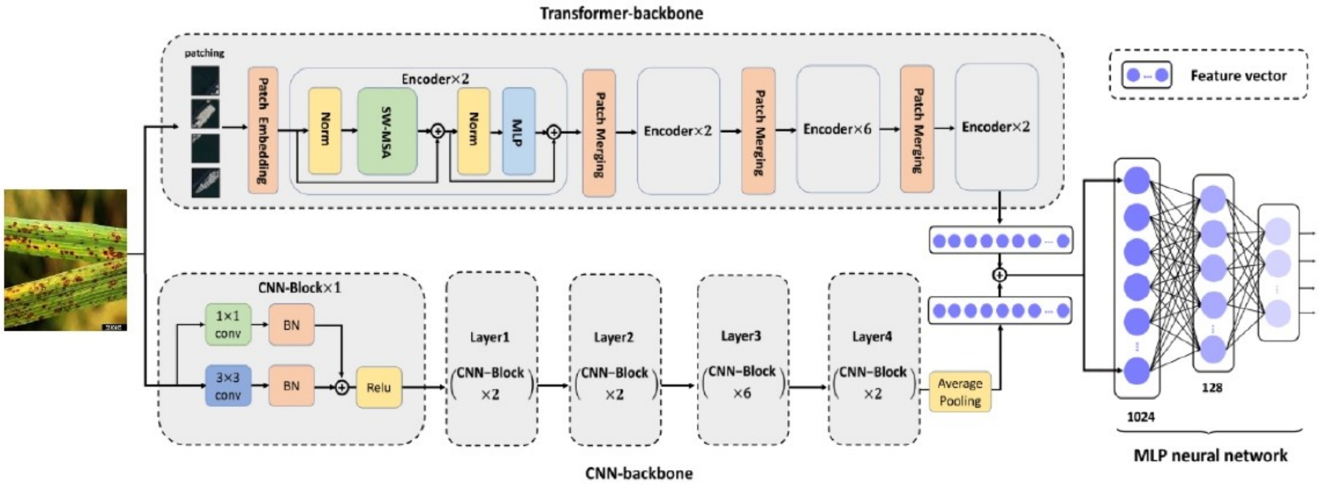


Fig. 2: Network architecture of the proposed model [2].

IV. RICE DISEASE TYPES AND DATASET DESCRIPTION

This experiment utilized a rice leaf disease dataset containing 18,445 images obtained from the reliable source Kaggle. The dataset encompassed ten classes: bacterial leaf blight(0), brown spot(1), healthy leaves(2), leaf blast(3), leaf scald(4), narrow brown spot(5), neck blast(6), rice hispa(7), sheath blight(8), and tungro(9). A CNN technique was applied to this dataset and achieved the highest precision. The distribution of the data is visualized in a seaborn chart.

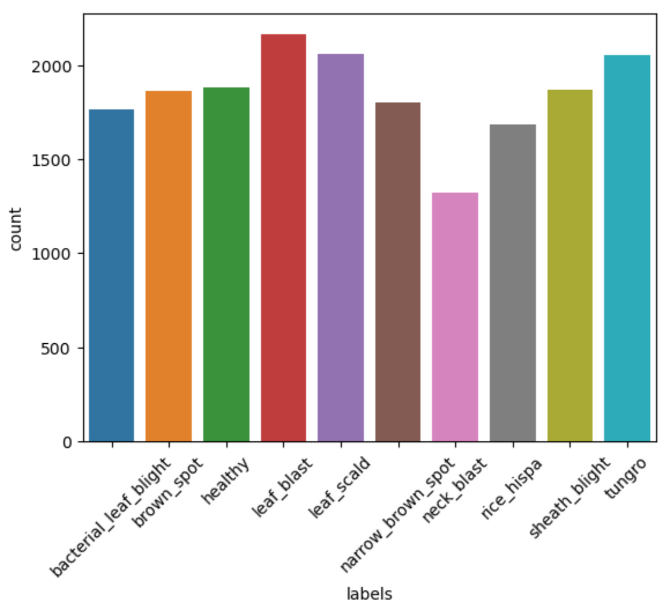


Fig. 3: Seaborn chart representation of rice leaf diseases dataset.

A. Bacterial_Leaf_Blight:

Xanthomonas oryzae is the causative agent of leaf blight, a bacterial disease. The diseased leaves curl up after being discolored, turning from green to a grayish tinge. Bacterial slime, which resembles drops of morning dew, is observed early in the formation of lesions. For an illustration of the effects of Leaf Blight on leaves, please see Figure 4(a).

Fig. 4: (a)bacterial_leaf_blight (b) brown_spot (c) healthy (d) leaf_blast (e)leaf_scald (f) narrow_brown_spot (g) neck_blast (h) rice_hispa (i) sheath_blight (j) tungro

B. Brown_Spot:

That's a fungal illness. The affected leaves may finally completely die as a result of several large spots. The leaves initially exhibit small, circular lesions that range in hue from dark brown to purple-brown.It has a reddish brown rim. The small dark brown lesions on the leaves affected by Brown Spot are seen in Figure 4(b).

C. Healthy_Leaves:

Healthy leaves are free from all types of diseases.We tested our model's ability to detect pictures free of illness using leaves from healthy rice plants. The corresponding image is displayed in figure 4(c).

D. Leaf_Blast:

Magnaporthe oryzae is the fungus that is causing the illness. Spindle-shaped or elliptical white to grey-green dots with dark red to brownish edges may be seen at first. Like a diamond, certain lesions may have wide centers and pointed ends. Figure 4(d) shows spindle-shaped lesions with dark brown edges and white spots.

E. Leaf_Scald:
The symptoms, which resemble greenish spore-like balls, appear on the panicle grains. These spore balls, which have a diameter of at least one centimeter, include whole floral components. The color eventually turns orange, then yellowish-green. Figure 4(e).

H. Rice_Hispa:
The rice hispa is one type of larval insect that defoliates plants. However, the rice hispa disease has not been properly researched in CNNs. In a recent paper, researcher [9] provided a method for clustering and classification using k-means, SVM, and CNN approaches. The study employed a dataset of 50 images of rice hispa with a 95% accuracy rate.

V. IMPLEMENTATION

A. Experimental Setup

The experiment was performed on a kaggle jupyter notebook equipped with GPU T4x2, 64-bit Operating System. The CNN-based model was implemented in the Keras 2.2.4 deep learning framework with TensorFlow 1.13.1 backend and python 3.7.2.

B. Image Acquisition

The images are collected from the reliable source kaggle as from internet. As discussed in the dataset description, data consists of 10 classes namely Bacterial_Leaf_Blight, Briwn_Spot, Healthy_Leaf, Leaf_Blast, Leaf_Scald, Narrow_Brown_Spot, Neck_Blast, Rice_Hispa, Sheath_Blight, Tungro.

C. CNN With Swin Transformer Model Training With Proposed Dataset

The model architecture used to categorize rice leaf infections is described in the table. Two distinct processing units receive the 224 × 224 pixel images: a Transformer and a Convolutional Neural Network (CNN). They all take useful information from the picture at the same time. The model then applies a straightforward addition operation to each linked element in order to combine the information from the CNN and Transformer. After that, a three-layer Multi-Layer Perceptron (MLP) neural network receives this aggregated feature vector. The merged feature vector is sent to the input layer. In the buried layer, 128 neurons process the information through calculations. The output layer generates the final output, which includes likelihood probabilities related to various diseases, using a softmax.

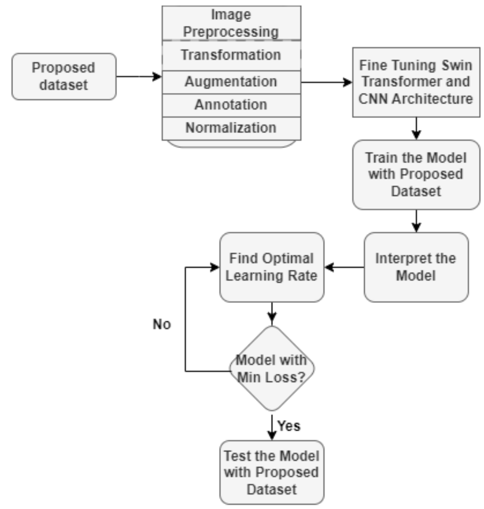


Fig. 5: Overview of the steps of proposed model.

D. VGG-16 CNN Model Training With Proposed Dataset

The training and testing process starts with the loading of the image data set. The relevant images and class labels are stored in separate arrays for training. To ensure a fair distribution, twenty percent of the data is utilized for training and the remaining eighty percent is reserved for testing. The train_test_split function is used for this.
Moreover, thirty percent of the eighty percent training data is reserved for validation when it is divided again. The class labels are encoded as numbers to speed up processing as much as feasible. After that, one-hot encoding is applied to these labels, converting each label's representation from an integer to a vector. Next, the keras library is used to import the VGG-16 model. The residual layers of the model are the ones that remain after the final completely linked layers are eliminated. After then, the remaining layers are designated as untrainable. Following the flattening of the feature extractor component's output, an output layer with softmax activation and a fully linked layer emerge. The model is constructed using the Adam optimizer with categorical_crossentropy as the classification loss function in order to get it ready for training. Fifty epochs of training are conducted, and it is concluded that the outcomes have stabilized at that point. The steps involved in the categorization process are shown in Figure 6.

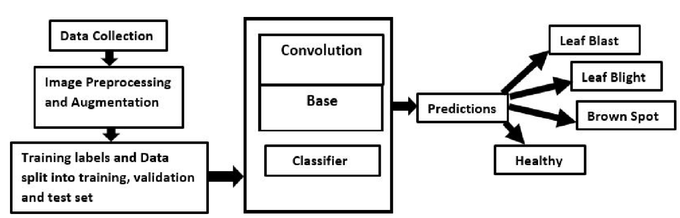


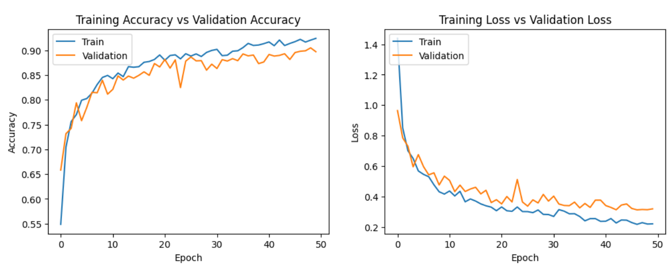
Fig. 6: Overview of the steps of Pre-trained VGG-16 model.

VI. Results and Performance Analysis

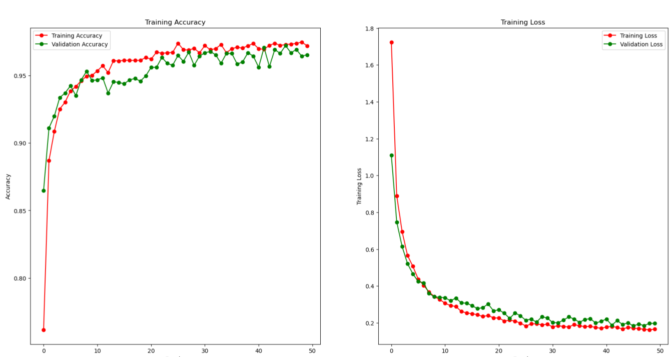
This section serves as an analog for the systematic execution of our suggested system. We have shown how we collected data, as well as identified the flaws that came from our exploratory data analysis and offered our proposed solutions. In addition, we implemented the base paper model with our proposed dataset. We also implemented our prosed model architecture with proposed dataset. Finally, we evaluated the proposed model and base paper model by using some measuring tools like accuracy, precision, recall, F1 score and confusion matrix.

A. Classification Results With Proposed Dataset

This portion shows and addresses the result derived by the proposed methodology and proposed dataset. A pre-trained VGG-16 with transfer learning and fine tuning CNN with swin transformer architecture were experimented with training and test sets containing 70% and 30% respectively (50 epochs). The existing paper architecture and proposed architecture with proposed datasets results are shown in the following tables and figures.



(a)



(b)

Fig. 7: Accuracy and Loss graph of training and validation for CNN model i.e, (a) Pre-trained VGG-16, (b) Fine tuning CNN with swin transformer architecture.

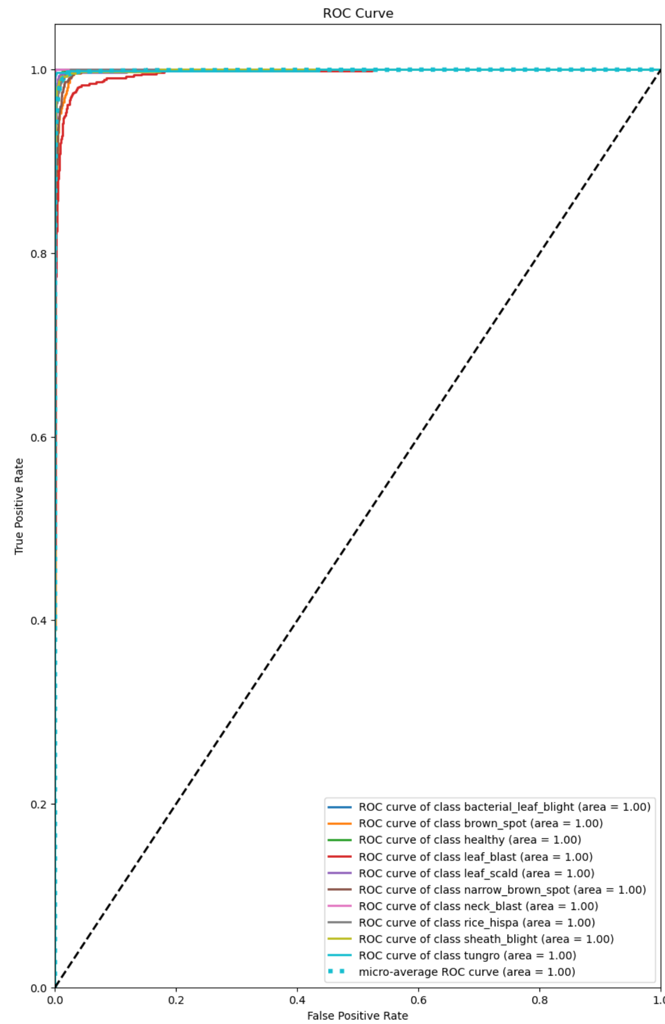
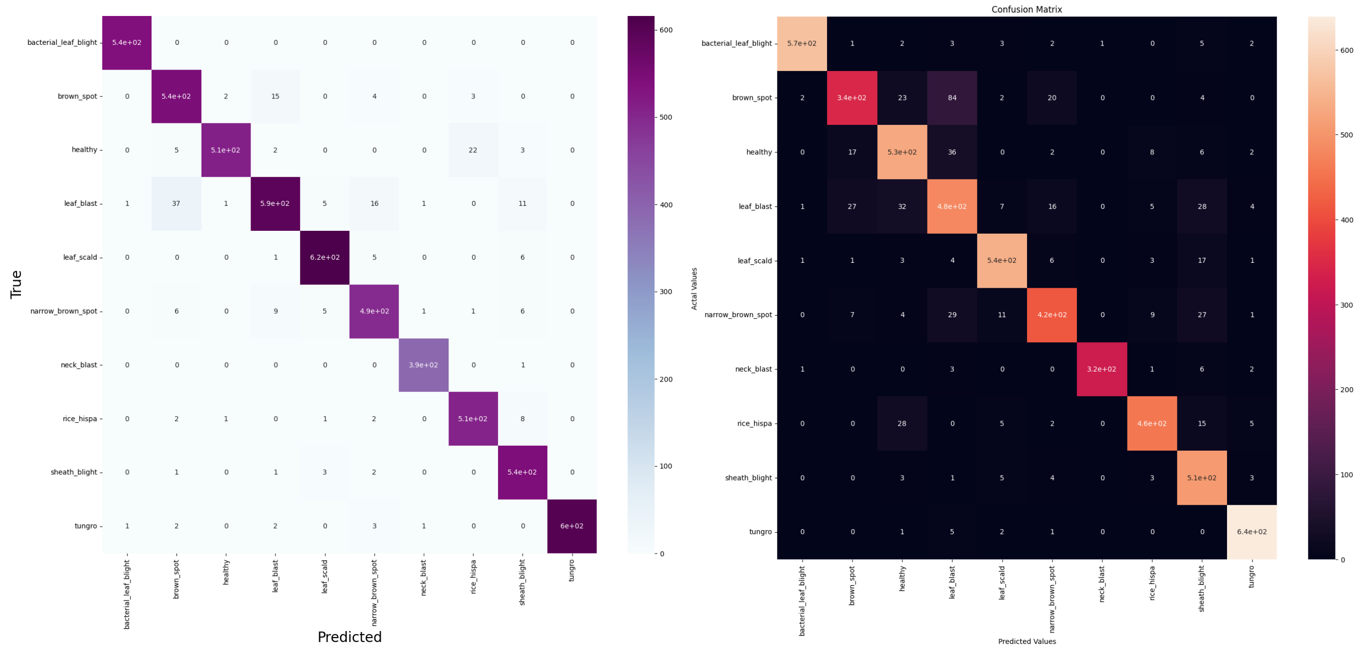


Fig. 8: ROC curve of proposed CNN and Swin transformer model.

TABLE I: Precision, Racall and F1-score of Pre-trained VGG-16 and CNN with swin transformer with Proposed Dataset.

Disease Leaves	Precision		Recall		F1-score		Support
Model	Swin	VGG-16	Swin	VGG-16	Swin	VGG-16	
0	1.00	0.99	1.00	0.97	1.00	0.98	586
1	0.91	0.87	0.96	0.72	0.93	0.79	480
2	0.99	0.85	0.94	0.88	0.97	0.86	600
3	0.95	0.74	0.89	0.80	0.92	0.77	600
4	0.98	0.94	0.98	0.94	0.98	0.94	576
5	0.94	0.89	0.95	0.83	0.94	0.86	504
6	0.99	1.00	1.00	0.96	0.99	0.98	336
7	0.95	0.94	0.97	0.89	0.96	0.92	512
8	0.94	0.82	0.99	0.96	0.96	0.89	528
9	1.00	0.97	0.99	0.99	0.99	0.98	648
accuracy					0.96	0.9183	5370
macro avg	0.97	0.90	0.97	0.89	0.97	0.90	5370
weighted avg	0.96	0.90	0.96	0.89	0.96	0.89	5370



(a) (b)

Fig. 9: Confusion Matrix for (a) Fine Tuning CNN with Swin Transformer Architecture.(b) Pre-trained VGG-16 with Proposed Dataset.

B. Model Comparison

TABLE II: Performance Comparison between existing and proposed model.

Criteria	Efficientnet (Bhujel et. al, 2022)	VGG-16	swin transformer
Train Accuracy	76.6%	92.42%	97.21%
Test Accuracy	85.12%	91.83%	96.40%
Train loss	0.6930	0.2208	0.1660
Validation loss	0.4456	0.3191	0.1970
Total parameters	156 M	211 M	1955 M

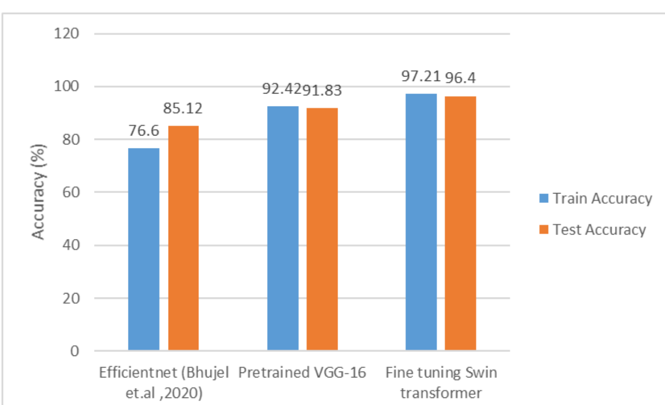


Fig. 10: Bar graph representation of existing model and proposed model.

C. Performance Analysis

Our study investigated the use of a pre-trained VGG-16 convolutional neural network with transfer learning and fine tuning CNN with Swin Transformer architecture for classifying rice leaf diseases. We employed a large dataset exceeding the size used in Shreya et. al (2020) [1]. Ten major rice plant diseases were included in the analysis. The results, visualized in the above Figures and tables demonstrate that the Swin Transformer achieved better performance. It reached a peak training accuracy of 97.21% and testing accuracy of 96.40% within 50 epochs. Swin transformer gave better result because their hierarchical processing approach allows them to effectively capture long-range dependencies within the images, which can be crucial for achieving better performance. However, this improvement comes at the cost of increased computational demands, as the Swin Transformer is a more intricate and parameter-intensive model compared to the pre-trained VGG-16 with transfer learning.

VII. CONCLUSION

This research investigates how dataset size and model architecture influence CNN performance for computer vision tasks. We emphasize data quality using Seaborn to ensure balanced classes. Data preprocessing guarantees consistent formatting and avoids compatibility issues, while augmentation expands the training data for robustness. Normalization prevents bias towards features with large initial values. Shreya Kamal et al. [1] achieved 92% accuracy on a small dataset (1649 images) using VGG-16, highlighting its limitations with restricted data. We implemented their VGG-16 architecture on a medium-sized dataset (18445 images) but saw minimal improvement, suggesting it struggles with larger datasets. To address this, we implemented a novel CNN-Swin Transformer hybrid model. This model achieved significantly better results (97.21% training accuracy, 96.40% testing accuracy) even with the larger dataset. The hybrid model integrates Swin Transformer blocks into the CNN backbone, enabling local feature extraction via CNN layers and global context capture through the attention mechanism of the Swin Transformer, potentially leading to superior performance. However, this increased accuracy comes at the cost of using more parameters compared to VGG-16, as Swin Transformers rely on an attention mechanism for long-range dependencies.

In future work, There are numerous possibilities to expand the scope of this study in the future. The main focus will be to evaluate the model performance by using dataset that contains large number of direct field images.

The main focus will be to evaluate the model performance by using dataset that contains large number of direct field images. The goal is to improve the accuracy and precision by exploring different configurations for the swin transformer blocks within the hybrid model. Experiment with the number of stages, transformer heads and window sizes to potentially improve performance.

If circumstances allow, we aspire to make a contribution to these challenges in the coming time, with the goal of enhancing the model to be more versatile and widely applicable than ever before.

References

[1] S. Ghosal and K. Sarkar, "Rice leaf diseases classification using cnn with transfer learn- ing," in 2020 IEEE Calcutta Conference (Calcon), pp. 230–236, IEEE, 2020.

[2] L. Huang, F. Wang, Y. Zhang, and Q. Xu, "Fine-grained ship classification by combining cnn and swin transformer," Remote Sensing, vol. 14, no. 13, p. 3087, 2022.

[3] V. K. Shrivastava, M. K. Pradhan, S. Minz, and M. P. Thakur, "Rice plant disease classification using transfer learning of deep convolution neural network," The Interna- tional Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 42, pp. 631–635, 2019.

[4] V. K. Shrivastava, M. K. Pradhan, and M. P. Thakur, "Application of pre-trained deep convolutional neural networks for rice plant disease classification," in 2021 international conference on artificial intelligence and smart systems (ICAIS), pp. 1023–1030, IEEE, 2021.

[5] . T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," arXiv preprint arXiv:1412.6806, 2014.

[6] S. Kaur, S. Pandey, and S. Goel, "Plants disease identification and classification through leaf images: A survey," Archives of Computational Methods in Engineering, vol. 26, pp. 507–530, 2019.

[7] S. S. H. Rumy, M. I. A. Hossain, F. Jahan, and T. Tanvin, "An iot based system with edge intelligence for rice leaf disease detection using machine learning," in 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1–6, IEEE, 2021.

[8] S. Lamba, A. Baliyan, and V. Kukreja, "A novel gcl hybrid classification model for paddy diseases," International Journal of Information Technology, vol. 15, no. 2, pp. 1127–1136, 2023.

[9] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural net- works," Biosystems Engineering, vol. 194, pp. 112–120, 2020.

[10] . Bhujel and S. Shakya, "Rice leaf diseases classification using discriminative fine tuning and clr on efficientnet," Journal of Soft Computing Paradigm, vol. 4, no. 3, pp. 172–187, 2020. 52

[11] F. Jubayer, J. A. Soeb, A. N. Mojumder, M. K. Paul, P. Barua, S. Kayshar, S. S. Akter, M. Rahman, and A. Islam, "Detection of mold on the food surface using yolov5," Current Research in Food Science, vol. 4, pp. 724–728, 2021.

[12] . E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, and M. Paul, "Rice leaf disease classi- fication and detection using yolov5," arXiv preprint arXiv:2209.01579, 2022.

[13] . Bala and V. Mehan, "Identification of rice plant diseases using image processing, ma- chine learning deep learning: a review," in CEUR workshop proceedings, vol. 3058, pp. 0–3, 2021.

[14] . Ahmed, T. R. Shahidi, S. M. I. Alam, and S. Momen, "Rice leaf disease detection using machine learning techniques," in 2019 International Conference on Sustainable Technolo- gies for Industry 4.0 (STI), pp. 1–5, IEEE, 2019.

[15] K. Kiratiratanapruk, P. Temniranrat, A. Kitvimonrat, W. Sinthupinyo, and S. Patara- puwadol, "Using deep learning techniques to detect rice diseases from images of rice fields," in International conference on industrial, engineering and other applications of applied intelligent systems, pp. 225–237, Springer, 2020.