2022 25th International Conference on Computer and Information Technology (ICCIT)
17-19 December, Cox's Bazar, Bangladesh

2022 25th International Conference on Computer and Information Technology (ICCIT) | 979-8-3503-4602-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCIT57492.2022.10055716

Generative Adversarial Network-based Augmented Rice Leaf Disease Detection using Deep Learning

Syed Taha Yeasin Ramadan[1], Tanjim Sakib[2], Md. Mizba Ul Haque[3], Nusrat Sharmin[4], Md. Mahbubur Rahman[5]
Department of Computer Science and Engineering
Military Institute of Science and Technology (MIST), Dhaka-1216, Bangladesh
Email: tahayeasin11@gmail.com, tsakib77@gmail.com, smmizba345@gmail.com, nusrat@cse.mist.ac.bd, mahbub@cse.mist.ac.bd

***Abstract*—Rice is one of the most produced crops in the world and the staple food in many South Asian countries. Rice leaf disease can affect the production of rice vastly, which can be prevented through the early detection of it. Many machine learning techniques have been used in recent years to help in the prevention of one of the most serious concerns, which is disease transmission.But there are limited images available of diseased leaf compared to healthy images which makes life tougher for machine learning models as they need a good amount of data for training. To solve this problem, a Generative adversarial network(GAN) has been used in recent days to create new, synthetic instances of an image that can pass as a real image. Recently, it has been used widely in the field of leaf disease identification. But there is very limited work done on rice diseases. In this paper, SRGAN (Super Resolution-GAN) has been considered as a data augmentation method to balance the dataset. Afterward, DenseNet121, DenseNet169, MobileNetV2, and VGG16 have been applied to classify the diseases. Experiment results show that the newly created augmented dataset produces the best results with both DenseNet169 and moboleNetV2 when compared to other models, with high accuracy of 94.30.**

***Index Terms*—Rice disease, Machine learning, Dataset, Data augmentation, GAN, SRGAN**

I. Introduction

Rice is the third most produced agricultural crop in the world, behind corn and sugarcane.For half of the world's population, rice is the main food source . Global rice production is expected to reach a record 509.9 million tons in 2021/22. The top countries to contribute in producing rice in the last year includes Australia, Bangladesh, China, Cote d'Ivoire, Cambodia, India, Laos, Indonesia, Nigeria, Paraguay, South Korea, Tanzania, Senegal, and Thailand. Among them Australia, Bangladesh, India, Thailand and China, has shown largest year to year increases. [1].

The current method primarily relies on developing seeds that are resistant to the majority of diseases and the manual farmer monitoring system. Since some areas are more susceptible to certain diseases than others, most diseases are prevented using area-based special procedures.However, in today's technological world, manual observation is insufficient for disease detection. It also takes a significant amount of expertise and hard labor, and there is always the possibility of human error in judgment. As a result, an automated system is required to address these concerns. Deep learning methods have been used in recent years to identify several leaf diseases, including rice [2] [3] [4].But the lack of images of infected plant leaves has long been the most significant barrier to plant disease detection.Some wide-spread diseases are presented in Fig. 1.

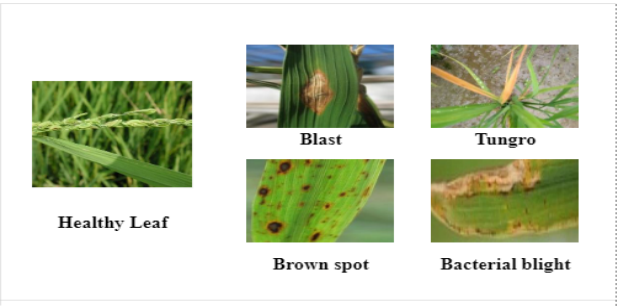

Fig. 1: Various Diseases of Rice Plant

In comparison to the millions of databases of healthy images, only a limited number of diseased photos are available. When deep learning models are trained on this unbalanced dataset, problems arise, and detection accuracy is poor.Generative adversarial Network(GAN) can help in this situation. To improve the training set, it is intended to produce or create entirely new synthetic pictures. The findings demonstrate that the GAN approach may significantly increase overall accuracy when compared to the traditional data augmentation techniques. It has lately seen extensive use in the detection of plant leaf diseases. To outperform earlier models, CNN models have been combined with several forms of GAN [5] [6].

When it comes to detecting rice diseases, use of GAN has been noticeably missing compared to other crops. GAN has shown promise in detecting other plant diseases, which cannot be overlooked. Recently, a work was done on a rice leaf dataset using StyleGAN2, and the author successfully demonstrated that using GAN with CNN models produces better outcomes than it did previously [7].

In this paper, SRGAN has been used as the model to augment the current dataset. This dataset was afterwards classified using four different types of deep learning architectures to determine which model produces the best results using SRGAN. The novelties of the paper are as follows

979-8-3503-4602-2/22/$31.00 ©2022 IEEE

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.

1) Using SRGAN as the data augmentation method for rice leaf disease and
2) Applying deep learning models to the augmented dataset to classify the rice leaf disease.

The remainder of the paper is organized as follows. Literature review is provided in Section 4. The methodology has been discussed in Section 4 and section 5 shows the experimental results, and the paper ends with a discussion of the experimental results and possible future study.

II. Related Works

People are using machine learning techniques recently to identify several leaf diseases, including rice. These approaches have successfully detected a variety of rice diseases while obtaining higher accuracy. In Paper [7], an Artificial Neural Network (ANN) was proposed as a classifier to distinguish between healthy and diseased leaves. It uses 300 photos and separates them into testing and training datasets. The accuracy of identifying infected leaves is 99 in training datasets but drops to 86 in testing. Paper [2] evaluated the performance of 11 CNN models. The deep feature of ResNet50 with SVM have the best result with an F1 score of 0.9838.

Wang Y, Wang H, and Peng Z. [3] developed a new deep learning model called ADSNN-BO and compared its performance to that of six other existing models. The results show that the newly built model exceeds the other existing models with a test accuracy of 96.65%. In Paper [4], a CNN model is trained using transfer learning, and the final training accuracy reached 90.9%.

A smartphone application has been developed that can identify diseased leaves after taking a picture using a camera. VGG16 is utilized as a classifier in the background and provides 100% train accuracy and an average test accuracy of 60% [8]. In paper [9], AlexNet was used to three distinct types of rice diseases—leaf smut, bacterial blight and brown spot—and achieved an impressive accuracy of 99%. Islam A [10] compares the performance of the three CNN models VGG, ResNet, and DenseNet on three different datasets, identifying ResNet50 as the worst one and DenseNet as the model with the best outcomes.In ref [11] an edge device was developed utilizing six different machine learning methods, including Random Forest [RF], Decision Trees [DT], Logistic Regression [LR], Support Vector Machine [SVM], K Nearest Neighbors [KNN] and Naive Bayes [NB]. Random Forrest method stands out as the best one after evaluation, with an average accuracy of 97.50%.

The paper [12] compares ANN and KNN-based classification methods for rice blast disease using a confusion matrix. The result shows that ANN-based classification methods outperform the other with 99% accuracy in detecting diseased leaves. A pre-trained model, InceptionResNetV2, was applied with a transfer learning strategy in ref [13]. The suggested model achieved a high accuracy of 95.67%.In paper [14], seven different classifiers, including classical and convolutional classifiers, are used to identify the fungus-causing blast disease. With a 73.12% accuracy rate, the random forest algorithm performs best for identifying the fungal blast disease. In paper [15],several machine learning algorithms, including KNN (K-Nearest Neighbor), J48 (Decision Tree), Logistic Regression and Naive Bayes were trained on the dataset. When applied to the test dataset, the decision tree method achieved an accuracy of 97% .

Generative Adversarial Network (GAN) is a relatively new data augmentation technique that has lately seen extensive use in the detection of plant leaf diseases. To outperform earlier models, CNN models have been combined with several forms of GAN [5] [6].However, very little research has been conducted on Rice utilizing the Generative Adversarial Network (GAN). Gugan Kathiresan [7] recently utilized GAN with the rice dataset and had success. He has been able to demonstrate that utilizing GAN has improved overall accuracy in comparison to using only CNN models alone. He combined StyleGAN2 with ResNet50. SRGAN has primarily been used in the past in the pharmaceutical sector, where high resolution images are required for various purposes. It has been used in [16]to convert images into high resolution because initially, getting high resolution images involves a lot of time because the patient must scan for a while, which causes discomfort.A similar method for MRI has been used in [17].

Fig. 2 displays a broad overview of the various machine learning methods employed in the field of rice disease detection.

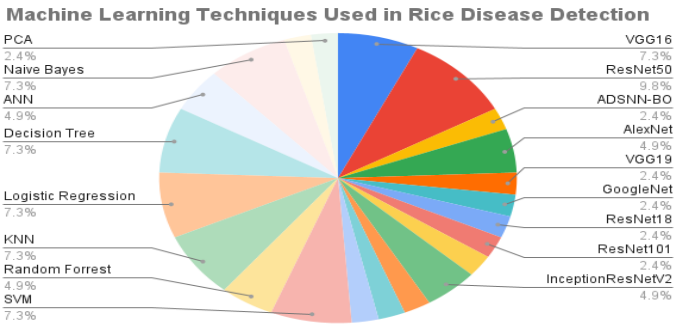

Fig. 2: Various machine learning methods used in rice disease detection

III. Methodology

We have divided the proposed methodology into two parts: i) Phase 1: Data Augmentation and ii) Phase 2: Classification. In phase 1 we applied SRGAN to augment the dataset and followed by the next phase we adopted four various deep learning models in order to classify the diseases.

A. Phase 1 Data Augmentation

SRGAN- Super Resolution Generative Adversarial Network: SRGAN also consists of two components: the generator and the discriminator.The generator generates data based on probability distributions, while the discriminator tries to predict whether input data came from the generator or the input

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.

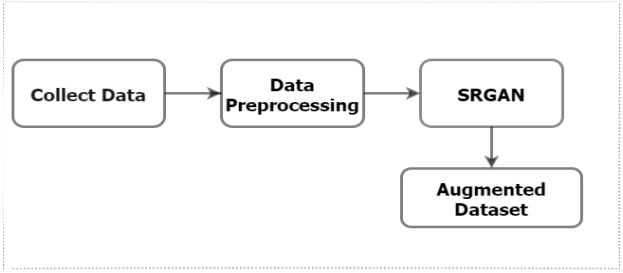

Fig. 3: Phase 1-Data Augmentation

dataset. But in this case, low resolution images are utilized rather than adding noise to the generator. The discriminator in this instance operates as usual. Another distinguishing feature of the SRGAN architecture is the usage of the perceptual loss function. This additional loss function, known as VGG loss, aids adversarial loss in producing superior results. The combined effects of the GAN loss and the content loss are very positive. The basic architecture of SRGAN is shown in figure 4.

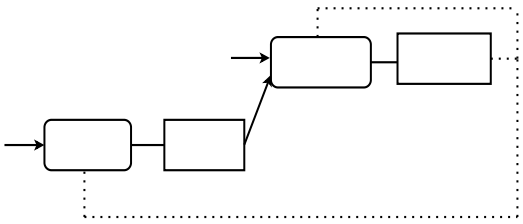

Fig. 4: General architecture of SRGAN

Generator that has been used here is using residual networks. The reason behind this is that they are easily trainable and also gives superior results compare to deep convolutional networks. It happens because of the use of skip connections.The model is shown in figure 5 The discriminator's job is

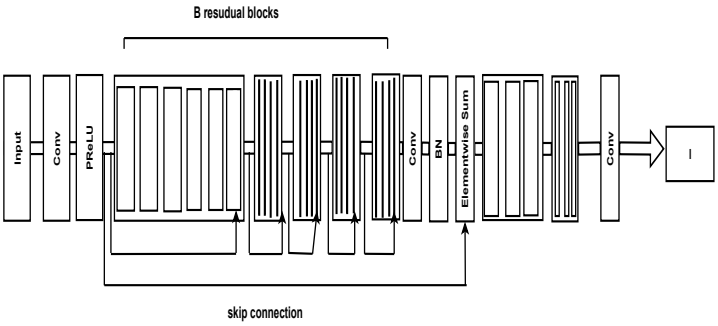

Fig. 5: Generator architecture of SRGAN

to differentiate actual High-resolution image and the produced super resolution image.It is quite similar to the architecture of DC-GAN .It has 8 convolutional layers with filter kernels of 3x3.To downgrade the image strided convolutions are used.The model is shown in figure 6.

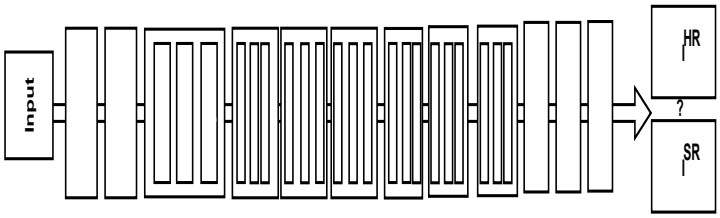

Fig. 6: Discriminator architecture of SRGAN

Fig. 3 depicts the phase 1 of data augmentation procedure. Data augmentation invreases the number of data by generating the data from the original dataset. Firstly, the rice leaf image dataset has been collected and then preprocessed the dataset. SRGAN has been applied to the preprocessed dataset for augmenting the dataset.

B. Phase 2 classification

To categorize the diseases, four deep learning models were used: DenseNet121, VGG16, MobileNetV2, and DenseNet169.

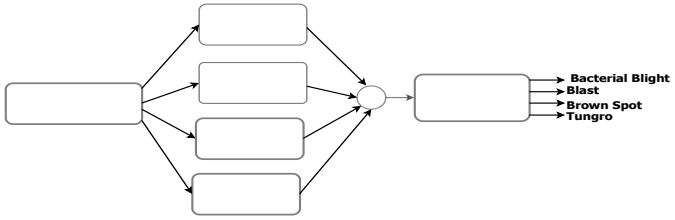

Fig. 7: Phase 2-Classification of the Diseases

Classification of rice leaf diseases has been portrayed in figure 7.

C. Implementation

A brief description of the steps is provided below.

- The dataset includes four categories of rice diseases: blast, brown spot, bacterial blight and tungro.
- Preprocessing includes cropping images otherwise the dataset would have different dimension images. Images have been rescaled to the same height and width.
- 128x128 images are used for high resolution and 32x32 images are used for low resolution images by downscaling the high-resolution images.
- From the low resolution images, the generator up-samples and creates SR (Super Resolution) images.
- As illustrated in fig. 4, a discriminator is utilised to separate the high resolution images, and the discriminator and generator are trained using back-propagation of the GAN loss.
- When the discriminator determines that the data is real, it is added to the dataset that should later be used for disease detection.
- CNN models DenseNet121, DenseNet169, MobileNetV2, and VGG16 are employed, and the results are compared to select the suitable model.

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.

IV. Experiments and Results

A. Dataset

We have conducted the experiment based on the dataset available from [18]. The images used here contain four different types of diseases. There are 5932 images in all. Table 1 provides a summary in tabular form. This initial dataset will be used to expand the number of images further.

TABLE I: The State of The Original Dataset

Disease	Number of samples
Bacterial Blight	1584
Blast	1440
Brown Spot	1600
Tungro	1308

B. Expanding the dataset using SRGAN

We have applied GAN to increase the dataset by nearly 33% and the details of the parameter in the generator, discriminator, SRGAN, VGG16 model have been reported in table II, III, IV, and V respectively.

Fig. 8 and 9, respectively, show the generator and discriminator loss states.

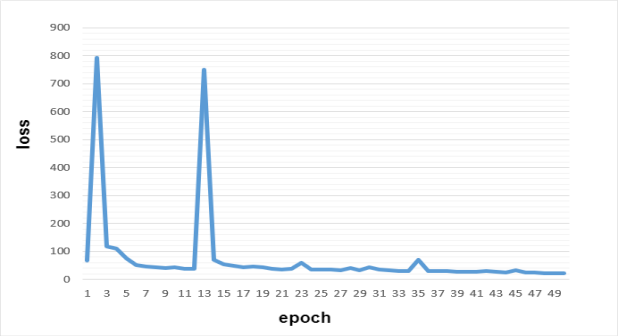

Fig. 8: Generator Loss

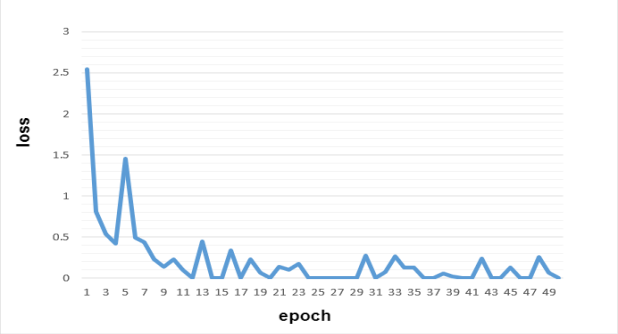

Fig. 9: Discriminator Loss

In Figure 10 shown a comparison among the low resolution, high resolution and GAN generated images. The leftmost image is the low resolution image produced by the same high resolution image that is seen in the figure's rightmost image.

TABLE II: PARAMETER OF GENERATOR MODEL

Layer (type)	Output Shape	Parameter
input_1 (InputLayer)	[(None, 32, 32, 3)]	0
conv2d (Conv2D)	(None, 32, 32, 64)	15616
p_re_lu (PReLU)	(None, 32, 32, 64)	64
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization	(None, 32, 32, 64)	256
p_re_lu_1 (PReLU)	(None, 32, 32, 64)	64
conv2d_2 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_1	(None, 32, 32, 64)	256
add (Add)	(None, 32, 32, 64)	36928
batch_normalization_2	(None, 32, 32, 64)	256
p_re_lu_2 (PReLU)	(None, 32, 32, 64)	64
conv2d_4 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_3	(None, 32, 32, 64)	256
add_1 (Add)	(None, 32, 32, 64)	0
conv2d_5 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_4	(None, 32, 32, 64)	256
p_re_lu_3 (PReLU)	(None, 32, 32, 64)	64
conv2d_6 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_5	(None, 32, 32, 64)	256
add_2 (Add)	(None, 32, 32, 64)	0
conv2d_7 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_6	(None, 32, 32, 64)	256
p_re_lu_4 (PReLU)	(None, 32, 32, 64)	64
conv2d_8 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_7	(None, 32, 32, 64)	256
add_3 (Add)	(None, 32, 32, 64)	0
conv2d_9 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_8	(None, 32, 32, 64)	256
p_re_lu_5 (PReLU)	(None, 32, 32, 64)	64
conv2d_10 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_9	(None, 32, 32, 64)	256
add_4 (Add)	(None, 32, 32, 64)	0
conv2d_11 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_10	(None, 32, 32, 64)	256
p_re_lu_6 (PReLU)	(None, 32, 32, 64)	64
conv2d_12 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_11	(None, 32, 32, 64)	256
add_5 (Add)	(None, 32, 32, 64)	0
conv2d_13 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_12	(None, 32, 32, 64)	256
p_re_lu_7 (PReLU)	(None, 32, 32, 64)	64
conv2d_14 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_13	(None, 32, 32, 64)	256
add_6 (Add)	(None, 32, 32, 64)	0
conv2d_15 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_14	(None, 32, 32, 64)	256
p_re_lu_8 (PReLU)	(None, 32, 32, 64)	64
conv2d_16 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_15	(None, 32, 32, 64)	256
add_7 (Add)	(None, 32, 32, 64)	0
conv2d_17 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_16	(None, 32, 32, 64)	256
p_re_lu_9 (PReLU)	(None, 32, 32, 64)	64
conv2d_18 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_17	(None, 32, 32, 64)	256
add_8 (Add)	(None, 32, 32, 64)	0
conv2d_19 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_18	(None, 32, 32, 64)	256
p_re_lu_10 (PReLU)	(None, 32, 32, 64)	64
conv2d_20 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_19	(None, 32, 32, 64)	256
add_9 (Add)	(None, 32, 32, 64)	0
conv2d_21 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_20	(None, 32, 32, 64)	256
p_re_lu_11 (PReLU)	(None, 32, 32, 64)	64
conv2d_22 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_21	(None, 32, 32, 64)	256

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.

Layer (type)	Output Shape	Parameter
batch_normalization_23	(None, 32, 32, 64)	256
add_11 (Add)	(None, 32, 32, 64)	0
conv2d_25 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_24	(None, 32, 32, 64)	256
p_re_lu_13 (PReLU)	(None, 32, 32, 64)	64
conv2d_26 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_25	(None, 32, 32, 64)	256
add_12 (Add)	(None, 32, 32, 64)	0
conv2d_27 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_26	(None, 32, 32, 64)	256
p_re_lu_14 (PReLU)	(None, 32, 32, 64)	0
conv2d_28 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_27	(None, 32, 32, 64)	256
add_13 (Add)	(None, 32, 32, 64)	64
conv2d_29 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_28	(None, 32, 32, 64)	256
p_re_lu_15 (PReLU)	(None, 32, 32, 64)	0
conv2d_30 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_29	(None, 32, 32, 64)	256
add_14 (Add)	(None, 32, 32, 64)	64
conv2d_31 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_30	(None, 32, 32, 64)	256
p_re_lu_16 (PReLU)	(None, 32, 32, 64)	64 36928
conv2d_32 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_31	(None, 32, 32, 64)	256
add_15 (Add)	(None, 32, 32, 64)	0
conv2d_33 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_32	(None, 32, 32, 64)	256
add_16 (Add)	(None, 32, 32, 64)	0
conv2d_34 (Conv2D)	(None, 32, 32, 64)	147712
up_sampling2d (UpSampling2D)	(None, 32, 32, 64)	0
p_re_lu_17 (PReLU)	(None, 32, 32, 64)	256
conv2d_35 (Conv2D)	(None, 32, 32, 64)	590080
up_sampling2d_1 (UpSampling2D)	(None, 32, 32, 64)	0
p_re_lu_18 (PReLU	(None, 32, 32, 64)	256
conv2d_36 (Conv2D)	(None, 32, 32, 64)	62211
Total params	Trainable params	Non-trainable params
2,044,291	2,040,067	4,224

Fig. 10: Comparison among LR, SR and HR Image

The image created with SRGAN is in the middle and leftmost image is the actual high resolution image.

Fig. 11 depicts the dataset's augmentation scenario which has been generated by SRGAN. We have seen an sufficient increase of the dataset using GAN.

C. CNN model

After generating the dataset with the GAN, we have applied four different CNN model to identify the disease. The accuracy of detecting the diseases have been predicted using the CNN model Densenet121, VGG16, DenseNet169, and ResNet50. Table V report the parametr of VGG16 model. Due to the page limit we only present the parameter for one model. Table VI report the accuracy of each model.

The hyperparameters of the classifiers used to categorize the rice leaf disease are shown in Table VII.

TABLE III: PARAMETER OF DISCRIMINATOR MODEL

Layer (type)	Output Shape	Parameter
input_2(InputLayer)	[(None, 128, 128, 3)]	0
conv2d_37 (Conv2D)	(None, 128, 128, 64)	1792
leaky_re_lu (LeakyReLU)	(None, 128, 128, 64)	0
conv2d_38 (Conv2D)	(None, 64, 64, 64)	36928
batch_normalization_33	(None, 64, 64, 64)	256
leaky_re_lu_1 (LeakyReLU)	(None, 64, 64, 64)	0
conv2d_39 (Conv2D)	(None, 64, 64, 128)	73856
batch_normalization_34	(None, 64, 64, 128)	512
leaky_re_lu_2 (LeakyReLU)	(None, 64, 64, 128)	0
conv2d_40 (Conv2D)	(None, 32, 32, 128)	147584
batch_normalization_35	(None, 32, 32, 128)	512
leaky_re_lu_3 (LeakyReLU)	(None, 32, 32, 128)	0
conv2d_41 (Conv2D)	(None, 32, 32, 256)	295168
batch_normalization_36	(None, 32, 32, 256)	1024
leaky_re_lu_4 (LeakyReLU)	None, 32, 32, 256)	0
conv2d_42 (Conv2D)	(None, 16, 16, 256)	590080
batch_normalization_37	(None, 16, 16, 256)	1024
leaky_re_lu_5 (LeakyReLU)	(None, 16, 16, 256)	0
conv2d_43 (Conv2D)	(None, 16, 16, 512)	(None, 16, 16, 512)
batch_normalization_38	(None, 16, 16, 512)	2048
leaky_re_lu_6 (LeakyReLU)	None, 16, 16, 512	0
conv2d_44 (Conv2D)	(None, 8, 8, 512)	2359808
batch_normalization_39	(None, 8, 8, 512)	2048
leaky_re_lu_7 (LeakyReLU)	(None, 8, 8, 512)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 1024)	33555456
leaky_re_lu_8 (LeakyReLU)	(None, 1024)	0
dense_1 (Dense)	(None, 1)	1025
Total params	Trainable params	Non-trainable params
38,249,281	38,245,569	3,712

TABLE IV: PARAMETER OF SRGAN MODEL

Layer (type)	Output Shape	Parameter
input_1 (InputLayer)	[(None, 32, 32, 3)]	0
model (Functional)	(None, 128, 128, 3)	2044291
input_2 (InputLayer)	[(None, 128, 128, 3 0)]	0
model_1 (Functional)	(None, 1)	38249281
model_2 (Functional)	(None, 32, 32, 256)	2325568
Total params	Total params	Non-trainable params
42,619,140	2,040,067	40,579,073

The outcome shows that, among the MobileNetV2 and DenseNet169 are comparative better result with the accuracy of 94.30%, which is slightly higher than DenseNet121's accuracy. Note that all the results reported here are with the augment dataset through the SRGAN.

Figure 12 presents some examples of correctly and incorrectly identified images. As no model can provide 100% accurate results, there will be some instances in which it incorrectly

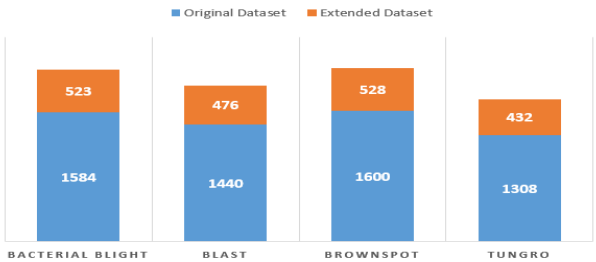

Fig. 11: Expanded dataset of rice leaf disease

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.

TABLE V: PARAMETER OF VGG16 MODEL

Layer (type)	Output Shape	Parameter
input_3 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv4 (Conv2D)	(None, 32, 32, 256)	590080
Total params	Total params	Non-trainable params
2,325,568	2,325,568	0

TABLE VI: The Result Obtained by CNN Models

CNN Model	Accuracy(%)
DenseNet121	93.73
MobileNetV2	**94.30**
VGG16	74.86
DenseNet169	**94.30**

TABLE VII: Hyperparameters of Classifiers

Parameter	Detail
Learning Rate	0.0001
Epochs	50
Batch Size	64
Optimizer	SGD
Loss Function	categorical_crossentropy

classifies the diseases. Additional data pre-processing methods may be utilized to increase accuracy further.

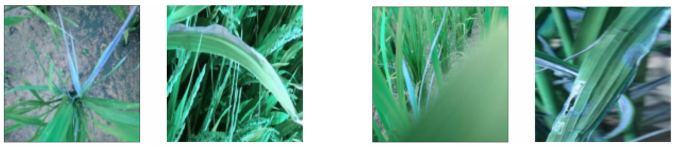

Fig. 12: Left two and right two images are correctly and incorrectly classified images respectively

V. Conclusion and Future Work

In this paper, we explore the use of GAN in the detection of rice diseases. GAN has been used in a substantial amount for leaf disease identification, for rice, it's yet to explore. In this work, we intend to apply the GAN with the popular CNN models. In this context, SRGAN was used in combination with four distinct deep-learning approaches. We have shown the result from the SRGAN with the increased number of the augmented dataset, With a training accuracy of 94.30, MobileNetV2 and DenseNet169 achieve the highest accuracy for this augmented dataset. To conclude, our proposed model able to detect various rice diseases with high accuracy, which will greatly benefit Bangladeshi farmers. However, we have applied only one kind of GAN here, which is the initial step of such research. In the future, other types of GANs can be utilized in rice disease datasets, as well as other key plants, to help in disease prevention.

References

[1] "Rice outlook: January 2022, author=Nathan W. Childs,Bonnie LeBeau, month=January, year=2022,."

[2] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, p. 105527, 2020.

[3] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention based neural network and bayesian optimization," *Expert Systems with Applications*, vol. 178, p. 114770, 2021.

[4] E. L. Mique Jr and T. D. Palaoag, "Rice pest and disease detection using convolutional neural network," in *Proceedings of the 2018 international conference on information science and system*, 2018, pp. 147–151.

[5] Y. Zhao, Z. Chen, X. Gao, W. Song, Q. Xiong, J. Hu, and Z. Zhang, "Plant disease detection using generated leaves based on doublegan," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[6] B. Nerkar and S. Talbar, "Cross-dataset learning for performance improvement of leaf disease detection using reinforced generative adversarial networks," *International Journal of Information Technology*, vol. 13, no. 6, pp. 2305–2312, 2021.

[7] G. Kathiresan, M. Anirudh, M. Nagharjun, and R. Karthik, "Disease detection in rice leaves using transfer learning techniques," in *Journal of Physics: Conference Series*, vol. 1911, no. 1. IOP Publishing, 2021, p. 012004.

[8] H. Andrianto, A. Faizal, F. Armandika *et al.*, "Smartphone application for deep learning-based rice plant disease detection," in *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*. IEEE, 2020, pp. 387–392.

[9] M. M. H. Matin, A. Khatun, M. G. Moazzam, M. S. Uddin *et al.*, "An efficient disease detection technique of rice leaf using alexnet," *Journal of Computer and Communications*, vol. 8, no. 12, p. 49, 2020.

[10] A. Islam, R. Islam, S. R. Haque, S. M. Islam, and M. A. I. Khan, "Rice leaf disease recognition using local threshold based segmentation and deep cnn," *Int. J. Intell. Syst. Appl*, vol. 13, no. 5, pp. 35–45, 2021.

[11] 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)*. IEEE, 2021, pp. 1–6.

[12] S. Ramesh and D. Vydeki, "Application of machine learning in detection of blast disease in south indian rice crops," *J. Phytol*, vol. 11, no. 1, pp. 31–37, 2019.

[13] N. Krishnamoorthy, L. N. Prasad, C. P. Kumar, B. Subedi, H. B. Abraha, and V. Sathishkumar, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, p. 111275, 2021.

[14] R. Kumar, G. Baloch, A. B. Buriro, J. Bhatti *et al.*, "Fungal blast disease detection in rice seed using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, 2021.

[15] K. 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 Technologies for Industry 4.0 (STI)*. IEEE, 2019, pp. 1–5.

[16] R. Sood, B. Topiwala, K. Choutagunta, R. Sood, and M. Rusu, "An application of generative adversarial networks for super resolution medical imaging," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 326–331.

[17] J. Liu, F. Chen, X. Wang, and H. Liao, "An edge enhanced srgan for mri super resolution in slice-selection direction," in *Multimodal Brain Image Analysis and Mathematical Foundations of Computational Anatomy*. Springer, 2019, pp. 12–20.

[18] P. K. Sethy, "Rice leaf disease image samples," *Mendeley Data*, vol. 1, 2020.

Authorized licensed use limited to: University Of Kalyani. Downloaded on August 13,2025 at 07:09:05 UTC from IEEE Xplore. Restrictions apply.