IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 December 2023, accepted 19 January 2024, date of publication 25 January 2024, date of current version 2 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3358333

RESEARCH ARTICLE

Real-Time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning

DIANA SUSAN JOSEPH[1], (Student Member, IEEE), **PRANAV M. PAWAR**[1], (Member, IEEE), AND **KAUSTUBH CHAKRADEO**[2]

[1]Birla Institute of Technology and Science Pilani, Dubai Campus, Dubai, United Arab Emirates
[2]Department of Public Health, University of Copenhagen, 1356 Copenhagen, Denmark

Corresponding author: Pranav M. Pawar (pranav@dubai.bits-pilani.ac.in)

ABSTRACT Agriculture plays a significant role in meeting food needs and providing food security for the increasingly growing global population, which has increased by 0.88% since 2022. Plant diseases can reduce food production and affect food security. Worldwide crop loss due to plant disease is estimated to be around 14.1%. The lack of proper identification of plant disease at the early stages of infection can result in inappropriate disease control measures. Therefore, the automatic identification and diagnosis of plant diseases are highly recommended. Lack of availability of large amounts of data that are not processed to a large extent is one of the main challenges in plant disease diagnosis. In the current manuscript, we developed datasets for food grains specifically for rice, wheat, and maize to address the identified challenges. The developed datasets consider the common diseases (two bacterial diseases and two fungal diseases of rice, four fungal diseases of maize, and four fungal diseases of wheat) that affect crop yields and cause damage to the whole plant. The datasets developed were applied to eight fine-tuned deep learning models with the same training hyperparameters. The experimental results based on eight fine-tuned deep learning models show that, while recognizing maize leaf diseases, the models Xception and MobileNet performed best with a testing accuracy of 0.9580 and 0.9464 respectively. Similarly, while recognizing the wheat leaf diseases, the models MobileNetV2 and MobileNet performed best with a testing accuracy of 0.9632 and 0.9628 respectively. The Xception and Inception V3 models performed best, with a testing accuracy of 0.9728 and 0.9620, respectively, for recognizing rice leaf diseases. The research also proposes a new convolutional neural network (CNN) model trained from scratch on all three food grain datasets developed. The proposed model performs well and shows a testing accuracy of 0.9704, 0.9706, and 0.9808 respectively on the maize, rice, and wheat datasets.

INDEX TERMS Deep learning, convolutional neural network, dataset development, plant disease classification, transfer learning.

I. INTRODUCTION

Plant diseases are among the leading causes of crop loss in many countries. Traditional disease analysis methods involve visual assessment by experts [1], [2]. However, it takes longer to diagnose the disease compared to automation techniques. Experts may only be available in some countries [3]. So, automatic recognition of plant diseases by image analysis is essential to solving this problem. Estimating the disease's severity is equally important to predict yield and suggest treatment [4].

Rice, wheat, and maize are important food grains with many health benefits. They are grown in large quantities and are energy giving food to the world. Food grains are an inevitable source of food for the growing population in many countries. These can be affected by a variety of

The associate editor coordinating the review of this manuscript and approving it for publication was Miaohui Wang.

diseases and pests. The diseases affecting them can further prevent the growth of the crop or affect the yield. Among the different food grains, the diseases affecting the rice, wheat, and maize/corn plants are given importance in this manuscript. Some common diseases affecting rice plants are Rice Blast, Sheath Blight, Bacterial Leaf Streak, Bacterial Leaf Blight, etc. [5]. The wheat plants' diseases include Powdery Mildew, Septoria Leaf Spot, Tanspot, Snowmold, etc. [6]. Charcoal rot, Downey mildew, Commonrust, black bundle disease, etc., are some diseases that commonly affect maize plants.

CNN [7], [8], [9] is a popular method that has been widely used in the field of plant disease detection [10], [11], [12]. The dataset used by the CNN models should be of good quality, and it has to be properly labeled. The performance of the CNN models greatly depends on the quality and amount of data used to train the model. The deep learning based CNN models require a large amount of data for training. It will not perform well if the model is trained using less data.

A dataset is an important component in machine learning or deep learning to learn and evaluate the performance of any algorithm or model. A dataset helps to organize the information collected from different sources and to arrive at the desired outcome. Only a well-organized dataset used for training a machine learning or deep learning model will be able to reflect the quality of the model and its effectiveness in arriving at the target outcome. One of the main challenges in plant disease diagnosis is the lack of availability of large and non-lab datasets. Some of the disadvantages of the existing datasets are most of the datasets available are laboratory processed; they include mostly images taken under controlled settings, a single leaf is mostly considered, they don't include all the symptoms of a particular disease, and complex backgrounds with varying lighting conditions are not considered. These disadvantages can limit the performance of disease detection by a CNN model, as the images can have multiple leaves with different complex backgrounds.

Our work focussed on these challenges, and three datasets for three important food grains, rice, wheat, and maize, were developed. The images in the datasets included all the disease stages early, advancing, and severe of all the diseases considered. The datasets includes processed images, images under complex backgrounds and lighting conditions, and images with multiple leaves under complex backgrounds. The images in the datasets were collected from internet sources and existing datasets. In the current work, the detection and classification of the diseases considered in the manuscript affecting rice, wheat, and maize were made using CNN models. Pre-trained eight CNN models trained on the ImageNet dataset were fine-tuned, trained, and tested on the datasets developed. A CNN model, namely Maize, Rice, Wheat-CNN model (MRW-CNN), is also proposed in our work to evaluate the performance of the developed datasets. Compared to the existing heavy state-of-the-art models for plant disease diagnosis, the MRW-CNN model has less number of layers, and the experimental results show that it generalizes well on the datasets. As a future enhancement, the model can be experimented with for disease diagnosis of other plant leaf datasets.

The main contributions of the current research work are as follows:

1) A real-time dataset for three important food grains, rice, wheat, and maize, was developed. Four diseases affecting each crop and a healthy class are present in all three datasets.
2) Early, advancing, and severe stages of the diseases were focused on while developing the datasets. Then, all three stages of a particular disease were put together in one class.
3) A hundred images were collected under each class for all five classes in a dataset. The collected images are pre-processed if needed and augmented to increase the size of the dataset.
4) A hundred images under each class of the datasets collected were annotated manually using the MakeSense AI tool for use in object detection models like Mask R-CNN.
5) Eight CNN models were fine-tuned, trained, and tested on the datasets developed. The performance of the fine-tuned CNN models and the proposed CNN model on the developed real-time datasets was evaluated using the accuracy and loss performance graphs, confusion matrices, classification reports, etc., on the test data.
6) A CNN model, namely MRW-CNN, is proposed for detecting maize, rice, and wheat leaf diseases. The proposed model can identify the features of the diseased and healthy maize, rice, and wheat leaves and accurately detect the diseases.
7) As a future extension of the current work, the different stages of the diseases included under a particular disease can be separated into classes like early, advancing, and severe, and predictions on disease stages can be made.

In the current work during the recognition of the maize leaf diseases, the models MobileNet, MobileNetV2, Inception V3, InceptionResNetV2, and Xception performed well with a testing accuracy of above 90%. For rice leaf disease recognition, all the models except ResNet 50 and ResNet 101 gave a testing accuracy above 90%. The models MobileNet, MobileNetV2, Inception V3, and InceptionResNetV2 gave a testing accuracy of above 90% on the wheat leaf dataset. The experimental results demonstrate that when trained from scratch on all the datasets developed, the proposed model gave a testing accuracy above 95%. The proposed model proved to generalize well on all three datasets considered in our work.

The rest of the paper is organized into seven sections. Section II deals with the related works in plant disease diagnosis. It focuses on the literature review of the works concentrating on dataset development for different plants and CNN models used for plant disease diagnosis. Section III deals with the steps in data set development, like data collection, pre-processing, augmentation, and annotation. Section IV discusses the development of the fine-tuned models, like how the CNN models selected are fine-tuned and trained, the basic architecture used, etc. Section V deals with the development of the proposed MRW model. Section VI deals with the results of the experiments conducted on the CNN models with the datasets developed for each food grain, experimental setup, hyperparameters used, visualization of activations, etc. Section VII discusses the case study of the application of the proposed method. Finally, Section VIII deals with the conclusion and future work.

II. RELATED WORKS

The review conducted in [13] mainly focuses on current developments in plant disease diagnosis using techniques like machine learning and deep learning and how these techniques speed up the process of disease detection in plants. The study elaborates on the importance of the detection of pests and diseases in plants, the various datasets used in plant disease detection, performance metrics used to evaluate the plant disease detection models, the challenges in the field, and future enhancements.

A 3DCNN is built to identify plant diseases in [14]; the work focussed on lesion segmentation and plant survival probability. The proposed model was evaluated on the diseases affecting three plants, pepper, potato, and tomato, from the Plant Village dataset. The model achieved a DICE coefficient of 90% and classification accuracies of 91.11%, 93.01%, and 99.04% for pepper, potato, and tomato plants, respectively.

A lightweight CNN model was developed in [15] based on VGG19 for disease diagnosis in peach leaves. Bacteriosis infection is mainly focussed on in the manuscript. The dataset for conducting the experiments was obtained from Farm of Agriculture University Peshawar, Pakistan. A total of 10,000 images were present in the dataset, and augmentation techniques were used to increase its size. The images in the dataset were also pre-processed.

The plant village dataset [16] contains 54,303 healthy and unhealthy images of leaves. The images in the dataset are laboratory processed. The plant village dataset is an open-access repository of images that can help to detect and classify plant diseases. The dataset is divided into 38 categories by species and diseases. Many studies in the field of plant disease diagnosis have made use of the plant village dataset [17], [18], [19], [20].

Temperature and carbon dioxide effect on maydis leaf blight disease affecting maize was analyzed in [21]. The results of the study showed that the severity of the disease can increase with the increase in temperature. Tolerance attributes were calculated for six disease stress based on diseased and non-diseased according to the yield. Wheat disease identification model based on EfficientNet architecture was proposed in [22] for detecting wheat rusts. A dataset consisting of 6556 images was prepared from natural field conditions. Several experiments were conducted using the CNN models VGG19, ResNet152, DenseNet169, Inception-NetV3, MobileNetV2, and eight variants of EfficientNet architecture; the results showed the fine-tuned EfficientNet B4 model proposed in the study performed better with a testing accuracy of 99.35% when compared to the other models.

Leaf tip detection and the number of leaves in a rice plant were determined by training a You Only Look Once (YOLO) deep learning algorithm in [23]. The study in [24] proposed a model that uses machine learning and deep learning advantages. The proposed model includes forty different hybrid deep learning models and five machine-learning techniques. The hyperparameters of the classifiers were optimized using the Optuna framework. A dataset for tomato early blight disease was collected from the Indian Agricultural Research Institute. The proposed model performed well on the dataset (IARI-TomEBD) formed and gave an accuracy of 87.55–100%. Validation of the model was also performed on PlantVillage-TomEBD and PlantVillage-BBLS.

A deep learning approach was proposed in [25] to identify diseases affecting maize crops. A dataset was prepared from fields of ICAR-IIMR, Ludhiana, India. Three different architectures based on the Inception-v3 framework were trained on the dataset prepared. The best performing model obtained an accuracy of 95.99%. A comparison of the state-of-the-art models with the model that performed best was done.

Analysis of Rice Blast disease by deep learning based model is done in [26] using remote sensing images. The proposed model obtained a training accuracy of 90.02% and a validation accuracy of 85.33%. Moderate resolution imaging spectroradiometer is used to establish the spatial distribution of leaf blasts. A deep learning based approach was used in [27] to detect tomato plant diseases. Two state-of-the-art semantic segmentation models, U-Net and Modified U-Net, were used to detect and segment the disease affected regions. Different U-Net versions were looked into for selecting the optimum one. Different classification methods were used in the study; both binary and multiclass classification of tomato diseases were done.

A dataset was developed by collecting 500 images of maize leaves from Google websites and the plant village dataset in [28]. The different periods of occurrence of maize diseases are included in the dataset. The size of the dataset was increased to 3060 images by applying augmentation techniques. There were eight categories of diseased leaves in the dataset. The study proposed two improved GoogLeNet and Cifar10 models for plant disease recognition. Eight kinds of maize diseases were identified.

A dataset of maize leaves was developed with the help of an Unmanned Aerial Vehicle (UAV) in [29]. The dataset contains 6267 images, where 3741 images are with lesions and 2526 are without lesions. The maize leaf images in the dataset are field images of the disease Northern Leaf Blight.

A CNN model ResNet-34 pre-trained on the ImageNet dataset was used in the study for training. The CNN model trained was used to generate heat maps that showed the disease regions.

The NLB dataset is used in [30]. It has three different parts. The first part is hand-held. The second part is the boom set. The last data set is an unmanned drone set. The hand-held part has more clarity; thus, the hand-held part of the dataset is used in this study. The dataset contains 1019 images with different angles and backgrounds and 7669 annotations. Augmentation techniques were applied to the dataset, and the size of the dataset was increased to 8152 images. A multi-scale feature fusion instance detection method was proposed, based on CNN (ResNet-101), to recognize the disease maize leaf blight.

A dataset of maize leaves was formed in [31] by collecting some images from the corn plantations in Raebareli and Sultanpur district, the rest from the Plant Village dataset. The dataset had three categories of images: Rust, Northern Leaf Blight, and Healthy. An Agriculture Scientist from India was consulted to label the images captured. A model was formed based on DCNN to identify two diseases affecting the maize plants.

The citrus dataset developed in [32] contains 759 images of leaves and citrus fruits, which are healthy and diseased. The diseases considered in the dataset are black spot, canker, scab, greening, and melanose affecting the citrus plants. The images in the dataset were manually captured using a DSLR with the help of an expert.

A real-time dataset of wheat diseases was developed in [33]. The dataset peers into most challenges for wheat disease diagnosis, specifically complex backgrounds, different capture conditions, etc. A model using deep learning and multiple instances learning for the real-time detection of wheat diseases was proposed in the study.

A dataset of wheat leaves was developed in [34] to detect the Fusarium Head Blight disease (FHB). The images in the dataset were collected from the field. Twelve images were collected initially, and later, it was increased to 2829 by applying data augmentation techniques. In this study, a method was proposed that uses Mask-RCNN along with color images to detect FHB.

The rice diseases image database was created with 500 common rice disease images of leaves and stems collected from the experimental field in [35]. This work trains the CNNs to identify ten diseases affecting rice plants. A real-time dataset of rice leaves was developed in [36]. The dataset contains a total of 8911 images. The rice leaf images included in the database contain both diseased and healthy images. The diseases considered are Rice blast, Red blight, Stripe blight, and Sheath blight. CNN was used for feature extraction, and d support vector machine (SVM) was used for predicting diseases.

PlantDoc, a Visual Plant Disease Detection dataset, was developed in [37]. The dataset contains 2,598 data points across 13 plant species. It includes 17 classes of diseases. The images were annotated using 300 human hours. The images in the dataset were downloaded from Google Images and Ecosia. Three models were used in the study to show the efficacy of the proposed dataset. An increase in classification accuracy by 31% was reported in their work.

From the literature survey conducted, it can be concluded that most of the datasets developed do not cover all of the underlying challenges existing in the field of plant disease diagnosis, like complex backgrounds, the variations of disease symptoms according to climate, the occurrence of multiple diseases on plants, etc. Only a few datasets are publicly available for further research or study. Some concentrate on the stages of the different diseases, but mainly, it is limited to one or two plants. Therefore, there is a need to develop a dataset that will address most of the underlying challenges in the field. The current manuscript mainly concentrates on developing a dataset of good quality for food grains rice, wheat, and maize. The dataset's primary focus is on real-life leaf images of the food grains considered. For example, Common Rust affecting the maize plants; for this particular disease, real-life images of all the stages are considered early, middle, and advancing and are included under CommonRust disease class, and the same for all diseases in the dataset developed.

III. PLANT DISEASE DATASET DEVELOPMENT

A real-time plant disease dataset for three important food grains, rice, wheat, and maize, based on leaf images, was developed in the current study. Diseases that cause great yield loss and have similar symptoms, many samples available, etc., were selected to form the dataset. Four diseases were selected for each food grain. For the diseases considered under each food grain, while collecting the images from internet sources, the leaf images for different stages of a particular disease, specifically early, advancing, and severe, were collected and put under the same disease class. More details on the dataset developed and the samples collected are available in Table 1, Table 2, and Table 3. In the current work, the diseased and healthy leaf images collected for each food grain are pre-processed, augmented, and then applied to deep learning based CNN models for detecting and classifying diseases. The results obtained are discussed in Section VI. Later, for the future extension of the work to be done, the images were manually annotated after the data collection and pre-processing steps for use in object detection models. Around 1500 images were annotated for all three food grains.

The overall process involved in developing the plant disease dataset for rice, wheat, and maize food grains in the current work is shown in Fig. 1. In Fig. 1, the different processes involved in developing the datasets are shown. The figure mainly shows the food grains and their stages selected in our work, the sources from which the datasets' images are collected, and the pre-processing and augmentation techniques used for processing the collected images. The different steps in developing the datasets are detailed in Subsections A, B, C, and D.

TABLE 1. Details of maize leaf disease dataset.

Class	Plant	Disease	Caused by	Type of disease	No of images	Areas affected	Occurrence
C1	Maize	Common Rust	Puccinia sorghi	Fungus	Total: 100 Early: 30 Advancing: 29 Severe: 41	Upper and lower leaf surfaces	Late May-Early July
C2		GreyLeafSpot	Cercospora zeae-maydis, Cercospora zeina.	Fungus	Total: 100 Early: 10 Advancing: 46 Severe: 44	Lower leaves are primarily affected	Every growing season
C3		Healthy	-	-	Total: 100	-	-
C4		Northern Leaf Blight	Exserohilum turcicum	Fungus	Total: 100 Early: 3 Advancing: 57 Severe: 40	Lower leaves are primarily affected	Relatively cool and wet seasons
C5		Southern Rust	Puccinia polysora	Fungus	Total: 100 Early: 17 Advancing: 41 Severe: 42	Primary infection on upper leaves	High humidity and temperatures above 80 F

TABLE 2. Details of rice leaf disease dataset.

Class	Plant	Disease	Caused by	Type of disease	No of images	Areas affected	Occurrence
C1	Rice	Rice Bacterial Leaf Streak	Xanthomonas oryzae pv. oryzicola	Bacteria	Total: 100 Early: 21 Advancing: 52 Severe: 27	Leaves	High temperature and high humidity
C2		Rice Bacterial Blight	Xanthomonas oryzae pv. oryzae.	Bacteria	Total: 100 Early: 8 Advancing: 51 Severe: 41	Leaves	Temperatures (25-34°C) relative humidity above 70%.
C3		Rice Blast	Magnaporthe oryzae	Fungus	Total: 100 Early: 21 Advancing: 39 Severe: 40	Leaves	Cooler temperatures
C4		Rice Brown Spot	Bipolarisoryzae	Fungus	Total: 100 Early: 4 Advancing: 65 Severe: 31	Leaves	High relative humidity (86-100%) and temperature between 16 and 36°C.
C5		Healthy	-	-	Total: 100	-	-

TABLE 3. Details of wheat leaf disease dataset.

Class	Plant	Disease	Caused by	Type of disease	No of images	Areas affected	Occurrence
C1	Wheat	LeafRust	Puccinia triticina	Fungus	Total: 100 Early: 10 Advancing: 36 Severe: 54	Primary infection on lower leaves	Moderate nights and warm days
C2		Powderymildew	Blumeria graminis f. sp. tritici	Fungus	Total:100 Early: 6 Advancing: 45 Severe: 49	Most prevalent on lower leaves	High humidity and temperatures (15-22C)
C3		StripeRust	Puccinia striiformis	Fungus	Total:100 Early: 9 Advancing: 46 Severe: 46	Leaves	Preferred by temperature (2-23C) and nonlimiting moisture
C4		Tanspot	Pyrenophora tritici-repentis	Fungus	Total:100 Early: 9 Advancing: 62 Severe: 29	Upper and lower leaves	
C5		Healthy	-	-	Total:100	-	-

A. DATA COLLECTION

The dataset includes real-time images collected from Google websites, Ecosia, Bing images, Flickr, etc. The images were manually selected and filtered. The dataset also comprises a few processed images for the healthy maize, rice, and wheat classes. Few images were taken from the plant village [16]

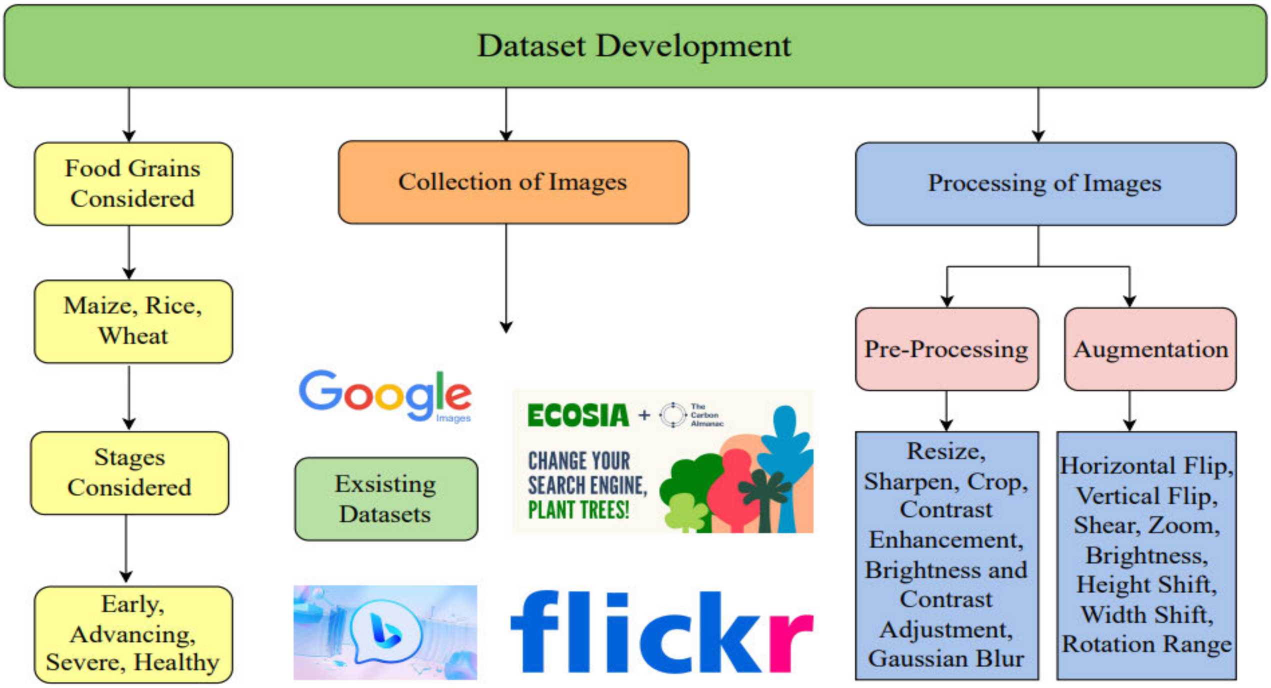

Dataset Development
Food Grains Considered
Maize, Rice, Wheat
Stages Considered
Early, Advancing, Severe, Healthy
Collection of Images
Google Images
Exsisting Datasets
ECOSIA + The Carbon Almanac
CHANGE YOUR SEARCH ENGINE, PLANT TREES!
flickr
Processing of Images
Pre-Processing
Augmentation
Resize, Sharpen, Crop, Contrast Enhancement, Brightness and Contrast Adjustment, Gaussian Blur
Horizontal Flip, Vertical Flip, Shear, Zoom, Brightness, Height Shift, Width Shift, Rotation Range

FIGURE 1. Dataset development methodology.

dataset for the healthy class of maize, and the healthy class of rice and wheat images were downloaded from the Kaggle website [38], [39]. The dataset was created by downloading images from the internet. A total of 5 classes were considered for each plant, including the healthy class, a different class from the diseased class. For all the diseased classes, images for all the stages of the disease were attempted to be collected, like images of early infection, advancing of the disease stage, and finally, the severe stage.

CNN models trained on real-life images are believed to perform better when compared to laboratory processed images. Therefore real-life images were given more importance while developing the dataset when compared to processed images. The crops considered for developing the dataset are rice, wheat, and maize. Four diseases affecting each of these plants were assessed for developing the dataset. The diseases considered for rice plants are Rice Bacterial Blight, Rice Bacterial Leaf Streak, Rice Blast, and Rice Brown Spot.

Similarly, the diseases considered for the wheat plant are Leaf Rust, Powdery Mildew, Stripe Rust, and Tan Spot. The diseases looked into for maize plants are Common Rust, Grey Leaf Spot, Northern Leaf Blight, and Southern Rust. Table 1, Table 2, and Table 3 give detailed descriptions regarding the type of disease, the number of images collected under each class and the number of images collected for early, advancing, and severe stages of the disease in each class, the areas of the leaves affected, and the conditions favorable for the disease to occur and advance; for the datasets developed for maize, rice, and wheat plants. The whole process of data collection is shown in Fig. 2. The samples of a few raw images collected for each food grain rice, wheat, and maize for the development of the datasets are shown in Fig. 3, Fig. 4, and Fig. 5.

B. DATA PRE-PROCESSING

Data pre-processing helps enhance some features of the image data required for processing. It can help in improving the accuracy and reliability of a dataset. Some images downloaded from the internet for the datasets were pre-processed before training the model. The software tools used for pre-processing the images were ImageJ [67], [68], and Python code. The purpose of pre-processing was to remove foreign objects from the images. The pre-processing steps applied to some images were resize, crop, sharpen, contrast enhancement, brightness and contrast adjustment, gaussian blur, etc. Python code was used to resize all the images in the dataset. The images were resized to 224 × 224. Some images were cropped using the ImageJ software and Python code to select the diseased areas of the leaves other than the background.

Some images in the dataset were manually selected and sharpened to visually focus on the diseased areas of the leaves more clearly. Similarly, other pre-processing techniques like contrast enhancement of the images to increase or decrease the contrast, brightness adjustments to increase or decrease the brightness of specific images, and gaussian blur to blur certain portions of the image were done using the

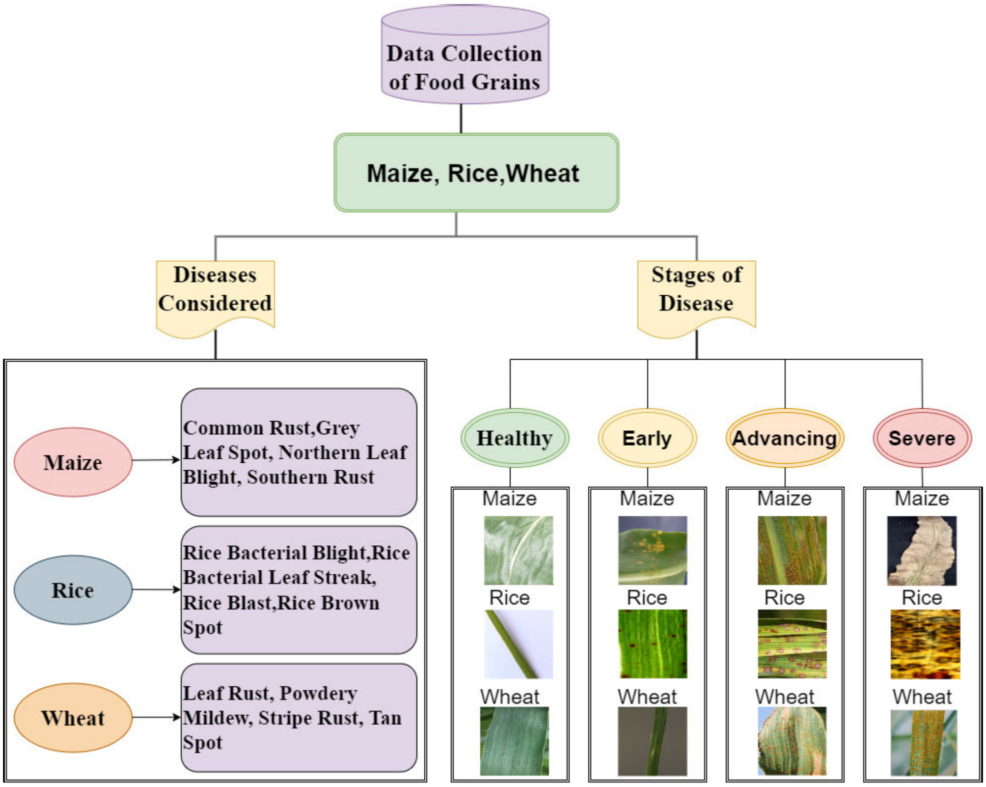

Data Collection of Food Grains
Maize, Rice,Wheat
Diseases Considered
Stages of Disease
Maize
Common Rust,Grey Leaf Spot, Northern Leaf Blight, Southern Rust
Rice
Rice Bacterial Blight,Rice Bacterial Leaf Streak, Rice Blast,Rice Brown Spot
Wheat
Leaf Rust, Powdery Mildew, Stripe Rust, Tan Spot
Healthy
Early
Advancing
Severe
Maize
Rice
Wheat

FIGURE 2. Data collection.

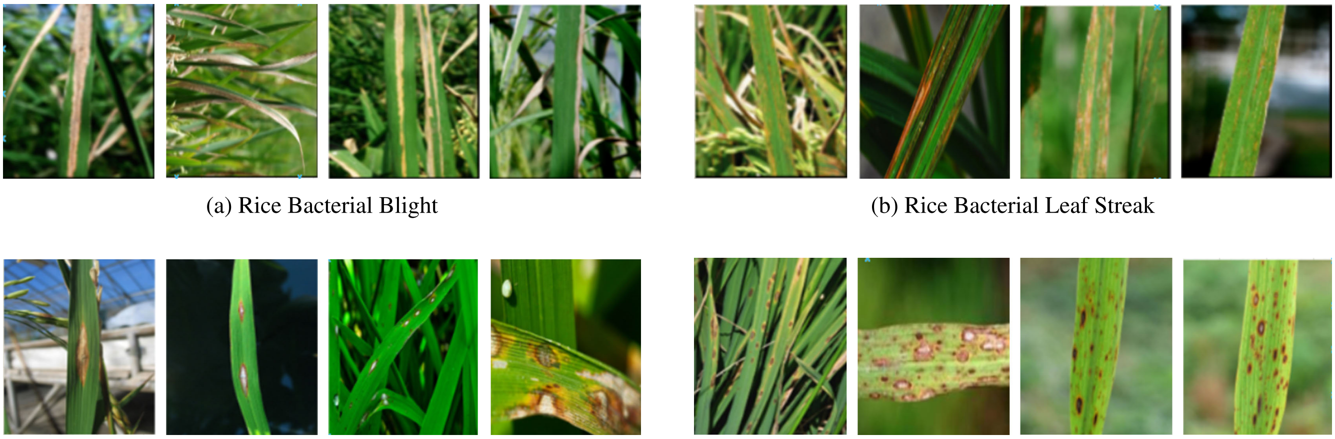


FIGURE 3. Samples of a few raw diseased images used to form the rice leaf dataset [40], [41].

ImageJ software. Examples of a few pre-processed images of Common Rust disease affecting the maize plant from the datasets are shown in Fig. 6.

C. DATA AUGMENTATION

A total of hundred images were collected under each class for the plants considered in this manuscript. The number of images under each class was insufficient to train a deep learning model. Therefore, data augmentation techniques were applied to the dataset to increase the size of the dataset. Python code was used to augment all the images in the dataset. Offline augmentation was done to all the images and saved to a folder. Then, the original images were combined with the augmented images of each class to form the dataset.

The images in all three datasets collected were augmented by applying specific augmentation techniques like horizontal and vertical flipping of the images, rotating the images by 90 degrees, applying a height shift range of 0.2, applying a width shift range of 0.2, zooming and shearing of the images by a range of 0.2, a brightness range of 0.2 to 0.8 was

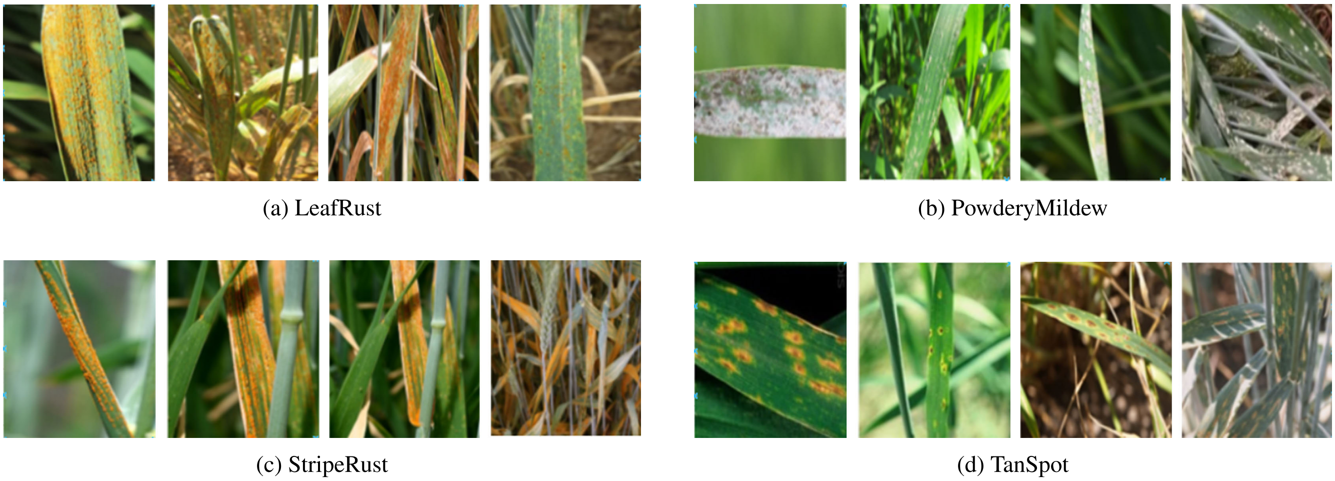


FIGURE 4. Samples of a few raw diseased images used to form the wheat leaf dataset [42], [43], [44], [45], [47], [48], [49], [50], [51], [52], [53].

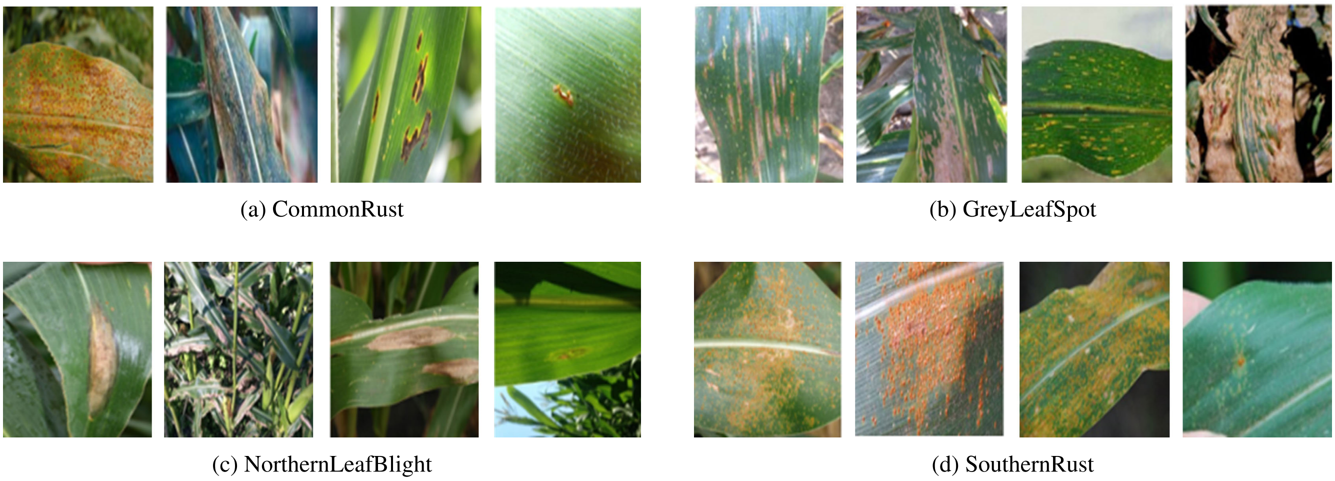


FIGURE 5. Samples of a few raw diseased images used to form the maize leaf dataset [54], [55], [56], [57], [59], [60], [61], [62], [63], [64], [65], [66].

applied to the images. Examples of a few augmented images of Tanspot and Powdery Mildew diseases from the datasets are shown in Fig. 6. Data augmentation techniques helped increase the model's ability to generalize and reduced the problem of overfitting.

D. DATA ANNOTATION

Data annotation can help to understand the meaning of data more precisely, increasing the performance of the deep learning models. The Make Sense AI [69] Tool was used in the current work to annotate the images in the dataset under each class. The Make Sense AI is an open-source and free to use annotation tool. It runs on a web browser. The different stages of a particular disease in the dataset were given importance during labeling. The whole process of data annotation is shown in Fig. 7. In the figure, the images of Common Rust disease affecting the maize plants are loaded into the Make Sense AI Tools web browser. The labels to be assigned to the images are mentioned. Each image uploaded is selected individually, and the polygon bounding box is selected to annotate the images. After annotating all the images uploaded to the web browser, the labels can be exported in YOLO, VOC, XML, VGG, JSON, and CSV formats. Since the polygon bounding box was used for annotation, the labels were exported in VGG and JSON formats.

IV. DEVELOPMENT OF FINE-TUNED MODELS

The collected, pre-processed, and augmented datasets for each food grain are split 80% for training and 10% each for validation and testing. The datasets for each food grain had a total of around 5000 images under each class after augmentation. The size of all three datasets was increased from 1500 images to 25000 images after augmentation. After splitting the dataset into training, validation, and testing, each class contains 4000 images for training and 500 images

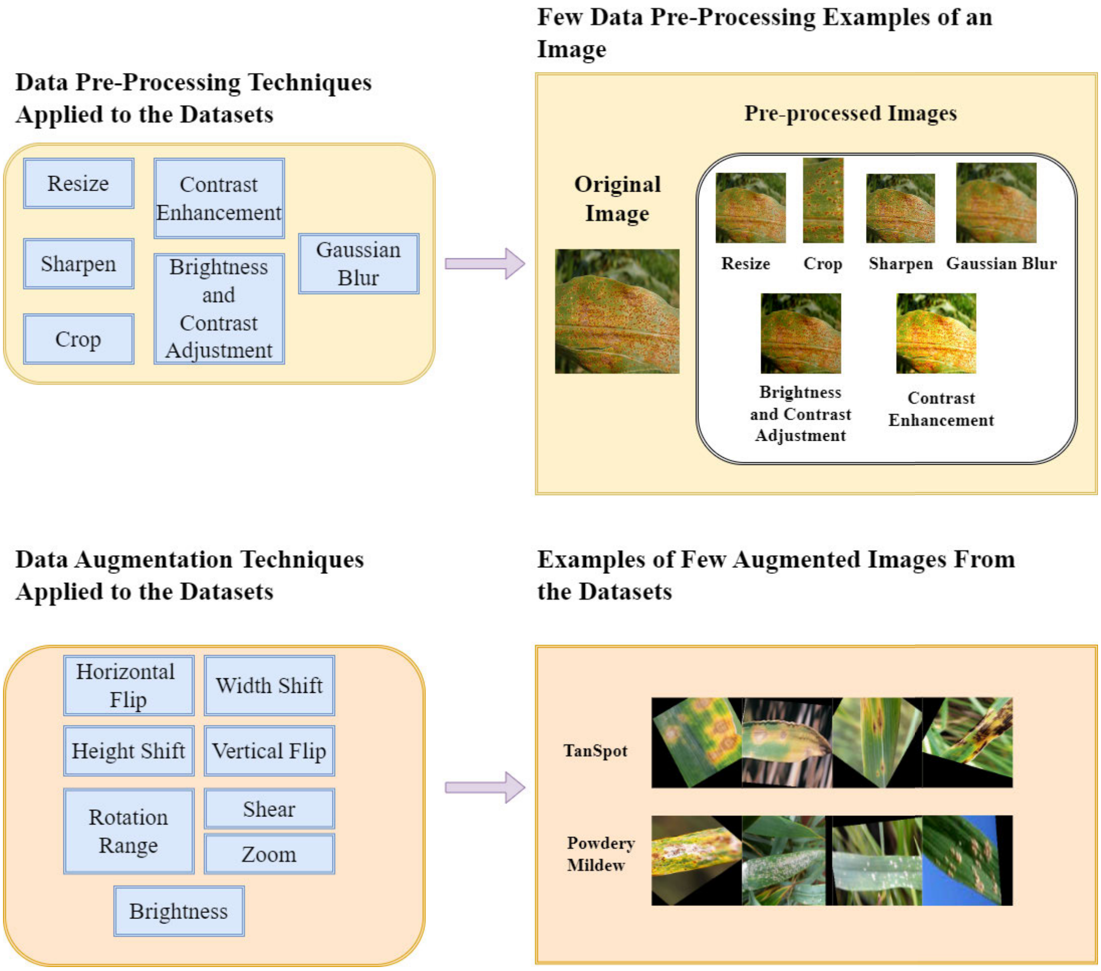

Data Pre-Processing Techniques Applied to the Datasets
Resize
Contrast Enhancement
Sharpen
Brightness and Contrast Adjustment
Gaussian Blur
Crop
Few Data Pre-Processing Examples of an Image
Pre-processed Images
Original Image
Resize
Crop
Sharpen
Gaussian Blur
Brightness and Contrast Adjustment
Contrast Enhancement
Data Augmentation Techniques Applied to the Datasets
Horizontal Flip
Width Shift
Height Shift
Vertical Flip
Rotation Range
Shear
Zoom
Brightness
Examples of Few Augmented Images From the Datasets
TanSpot
Powdery Mildew

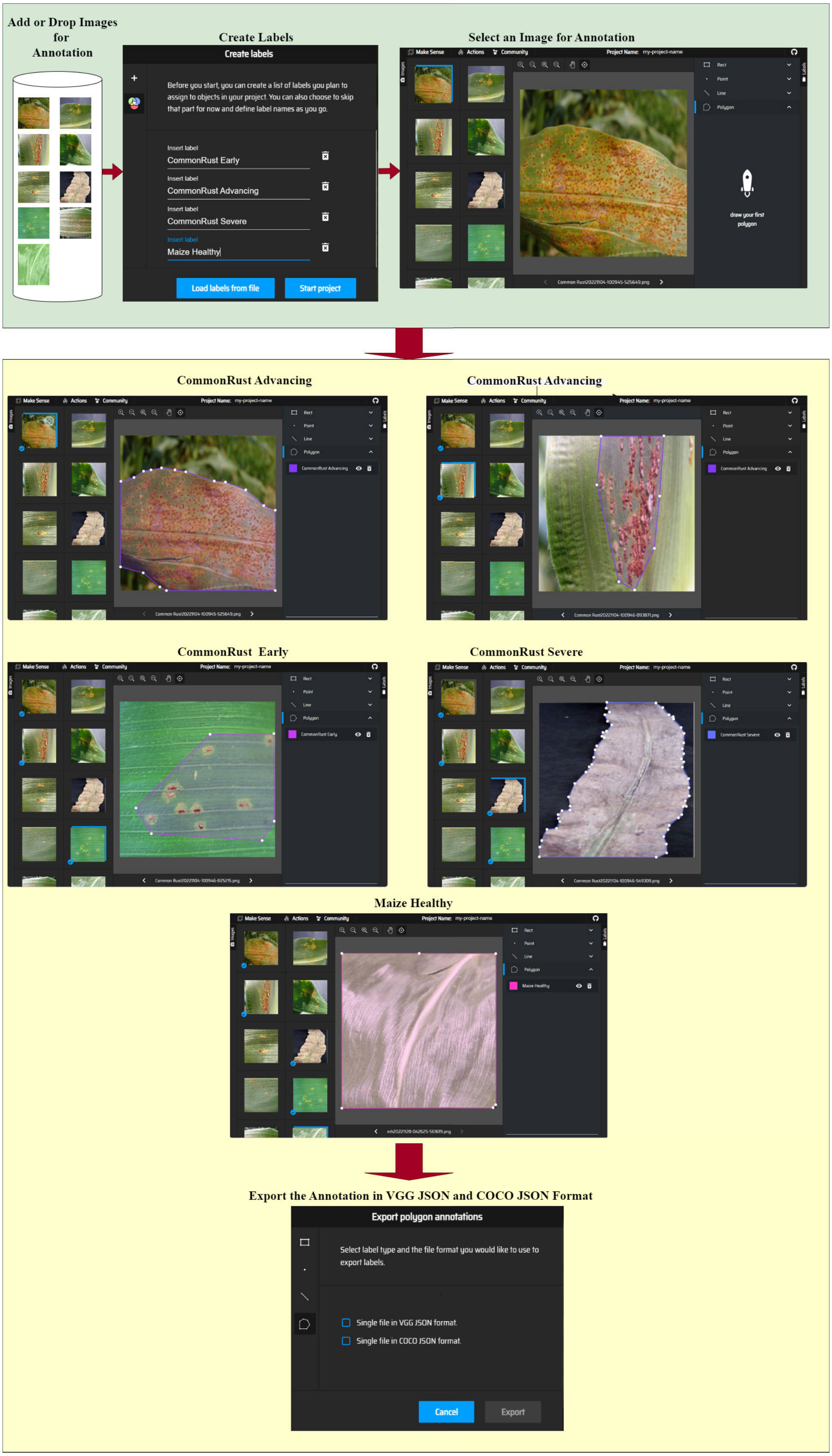
FIGURE 6. Data pre-processing and data augmentation.

Algorithm 1 Fine-Tuning of CNN Models

```
Input: Pre-trained models with top layers removed
Output: A fine-tuned model for detection and classification
        of diseases from the dataset developed
Load pre-trained models for fine-tuning
$m \leftarrow loadedmodel(pre - trained = True)$
while $m.toplayer = False$ do
    Set $layers.trainable = False$
    $x \leftarrow dropout(0.2)(m.output)$
    $x \leftarrow addBachNormalization(x)$ {x is output from previous layer}
    $x \leftarrow addFlattenlayer(x)$
    $x \leftarrow addDenselayer(x)$
    $l2regulizer \leftarrow 0.01$
    $x \leftarrow dropout(0.2)(x)$
    $predictions \leftarrow outputclasses(activation = softmax)$
    $finetunedmodel \leftarrow (m.input, predictions)$
end while
```

each for testing and validation. The number of images in all classes was balanced in all three datasets. The dataset developed for each plant was applied to various CNN based deep learning models like VGG16, ResNet 50, ResNet 101, Inception V3, InceptionResNetV2, MobileNetV2, and MobileNet. Pre-trained models trained on the ImageNet dataset [70] were downloaded from Keras and used. The concepts of transfer learning and fine-tuning were made use of for the training. Algorithm 1 discusses how the pre-trained models trained on the ImageNet dataset were fine-tuned. The steps in the algorithm include the following: the models were downloaded from Keras. The top layers from the eight models downloaded were removed. The layers trainable is set to False. After which, a few layers like Dropout, BachNormalization, Flatten, Dense layer, and the Softmax layer for the output classes were added.

A Dropout layer was added to nullify the effect of some neurons on the next layer. The BachNormalization layer works differently during training and inference. The layer can help the model to generalize well by normalizing its inputs in batches. The Bach Normalization layer can help smoothen the loss function by optimizing the model parameters and increasing the models' training speed. A Flatten layer was added to convert the inputs to one-dimensional. The Dense layer was added after the Flatten layer with a kernel-regularizer l2 of 0.01. Again, a Dropout of 0.2 was added.



Create labels

Before you start, you can create a list of labels you plan to assign to objects in your project. You can also choose to skip that part for now and define label names as you go.

Insert label: CommonRust Early
Insert label: CommonRust Advancing
Insert label: CommonRust Severe
Insert label: Maize Healthy

Export polygon annotations

Select label type and the file format you would like to use to export labels.

□ Single file in VGG JSON format.
□ Single file in COCO JSON format.

FIGURE 7. **Example of annotation of commonrust disease stages and healthy stage of maize using the makesense AI tool.**

The Softmax layer was added to get the output classes. Finally, a fine-tuned model was obtained where the input is the model's input, and the output is the predictions. After conducting several experiments, a Dropout of 0.2 and kernel-regularizer l2 and Bach Normalization layer, Early stopping was decided to be added to the pre-trained models to prevent over-fitting. Early Stopping automatically terminates the training when a performance measure does not improve further. The model architecture used is shown in Fig. 8.

Fig. 8 contains mainly two phases: the training and testing phases. In the training phase, the input images from either of the three datasets, which are pre-processed and augmented, are given as input to the state-of-the-art deep learning models. The datasets are split into 80% for training, 10% for validation, and 10% for testing before being given to the CNN models. The pre-trained models ResNet 50, ResNet 101, MobileNetV2, MobileNet, Inception V3, InceptionResNetV2, VGG 16, and Xception, trained on the ImageNet dataset, are selected in the current work for training on the datasets developed. The pre-trained CNN models selected are fine-tuned by removing the top layers and adding certain layers, as mentioned in Algorithm 2. The CNN models fine-tuned are trained on the training data and validated on the validation data. Examples of feature maps from a few layers of the InceptionResNetV2 model after training on the input images are also shown in Fig. 8.

In the testing phase, the trained CNN models are tested on the testing data, and predictions are made; the testing data is the data the model has not seen before. Each dataset developed for the food grains rice, wheat, and maize has five classes each. Therefore, a total of 15 output classes are shown in Fig. 8.

V. PROPOSED CNN MODEL OF MAIZE, RICE AND WHEAT (MRW-CNN) LEAF DISEASES

Fig. 9 shows the architecture of the proposed CNN model that has been used to classify the diseases affecting the food grains considered in our manuscript. The model consists of five Convolution layers, each followed by BatchNormalization and Rectified Linear Unit (ReLU). After the convolution layers, a Global Average Pooling layer is added, followed by a Dense layer with a Softmax activation function that acts as the output layer. The model is trained using Stochastic gradient descent (SGD). In the proposed CNN model, the convolution layers were used to extract the features from the training images; batch normalization is used to normalize the outputs from each layer during training for every batch. The layer helps to address the internal covariate shift problem. The activation function ReLU does not activate all the neurons at the same time; this helps to make the computation easier.

The problem of overfitting can be reduced by using the Global Average Pooling layer. The images can be down sampled using average pooling. The Dense layer is used for predicting the output class of the input image. The output of the Dense layer is used to evaluate the performance of the proposed model. The training and testing of the proposed CNN model was done by splitting the dataset developed for the three food grains into training, validation, and testing datasets. The datasets were split into 80% for training and 10% each for validation and testing. In the training phase, the input images from the training set that are pre-processed and augmented are given to the proposed model for training. The model is then validated on the validation dataset. In the testing phase, the trained MRW-CNN model is tested on the testing data which the model has not seen during training, predictions are then made. The training, validation, and testing datasets contain samples from all classes in the dataset. A total of five output classes are available for each of the datasets developed in our work.

VI. RESULTS

A. EXPERIMENT SETUP

Eight pre-trained models trained on the Imagenet dataset were downloaded from Keras and fine-tuned. GPU (NVIDIA ®Quadro ®P1000,4GB,4mDP) was used to train all fine-tuned models. Tensorflow, Keras, CUDNN, etc., were used for software implementation. The proposed MRW-CNN model was trained using the T4 GPU available in Google CoLab.

B. HYPERPARAMETERS

All the fine-tuned CNN based deep learning models are optimized by the SGD algorithm. The learning rate for all the models was set to 0.001, and a batch size 32 was chosen. Dropout operation was added to all the pre-trained models on the ImageNet dataset to prevent over-fitting and improve the generalization of the models. The MRW-CNN model is optimized by the SGD, a learning rate of 0.001 and batch size of 32 were used.

C. CONFUSION MATRIX

A confusion matrix can greatly help in visually estimating the performance of a model. The infected images of food grains considered in the current manuscript can be easily confused with multiple classes since most of the diseases selected for the development of the dataset have similar symptoms. The infected leaf images at different backgrounds can lead to confusion and complexity, which can result in performance deterioration.

The confusion matrix [71] is an m*m matrix that can be used to measure the performance of the classification models. In a confusion matrix, the X-axis contains the predicted values, and Y-axis includes the actual values. The True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) for each class in a multi-class classification can be computed from the confusion matrix, and from these, the values the precision, recall and F1-Score can be calculated per-class. The precision measures out of all predicted positives how many are actually positive. It can be

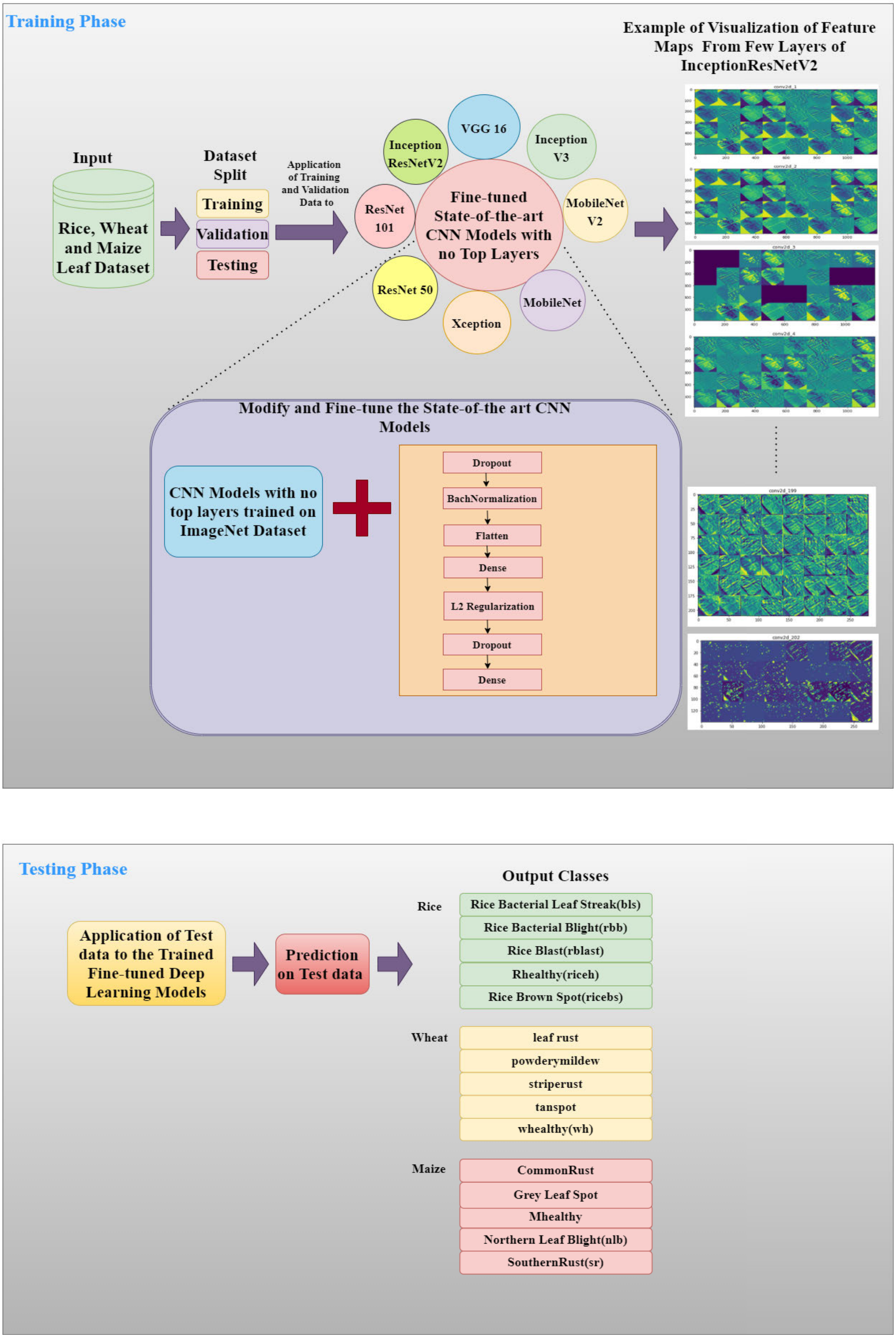


FIGURE 8. Architecture of Fine-tuned CNN Models.

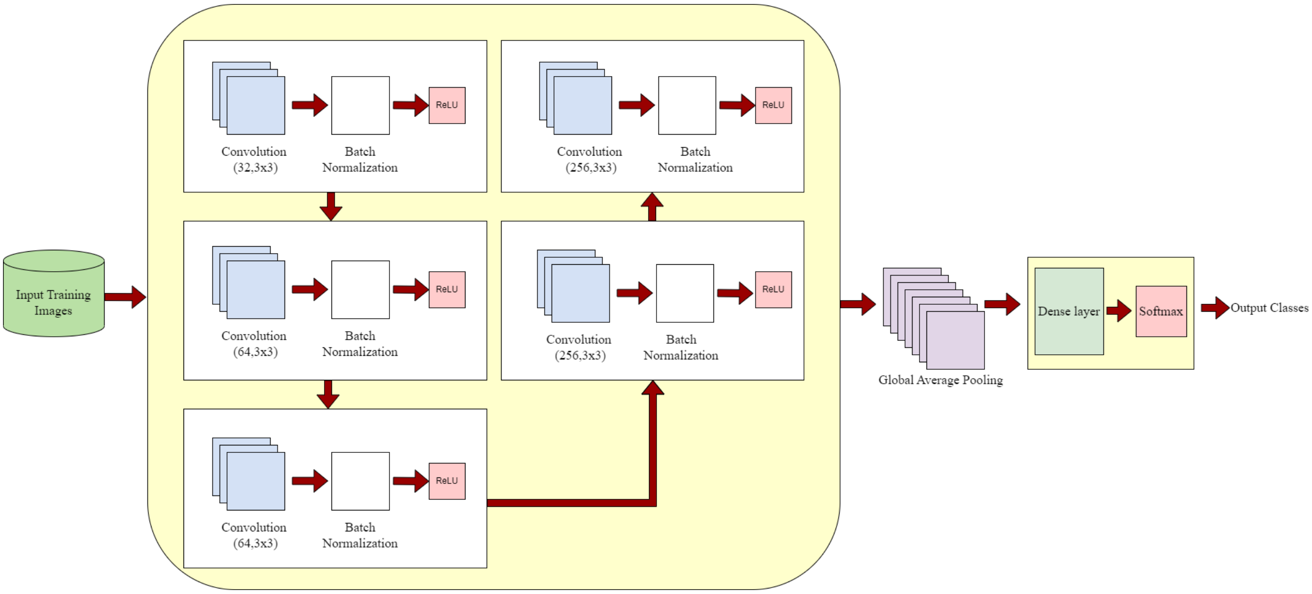


FIGURE 9. Architecture of proposed CNN model MRW-CNN.

calculated per-class using the equation (1)

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall measures how many positive records are predicted correctly. It can be calculated per-class using equation (2)

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1-Score is the harmonic mean of precision and recall it can be calculated per-class as expressed in equation (3)

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

D. CLASSIFICATION REPORT

The classification report is a performance evaluation metric that displays precision, recall,f1-score, accuracy, macro avg, weighted avg, and support for each class. It is used to measure the performance of the trained models on the test data. The accuracy is calculated by dividing the correct predictions by the total predictions from the confusion matrix obtained, as expressed in equation (4)

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} \tag{4}$$

Macro average recall scores are calculated by taking the arithmetic mean of all the per-class recall scores [72] as expressed in equation (5).

$$RecallMacroAvg = \sum_{i=1}^{n} \frac{Recall_i}{n} \tag{5}$$

Macro average precision scores are calculated by taking the arithmetic mean of all the per-class precision scores as expressed in equation (6)

$$PrecisionMacroAvg = \sum_{i=1}^{n} \frac{Precision_i}{n} \tag{6}$$

Macro average F1 scores are calculated by taking the arithmetic mean of all the per-class F1 scores as expressed in equation (7)

$$F1 - scoreMacroAvg = \sum_{i=1}^{n} \frac{F1 - score_i}{n} \tag{7}$$

The weighted-averaged F1-score [73] is calculated using equation (8) and equation (9) by taking the mean of all per-class F1 scores for a N-class dataset, while considering each class's support.

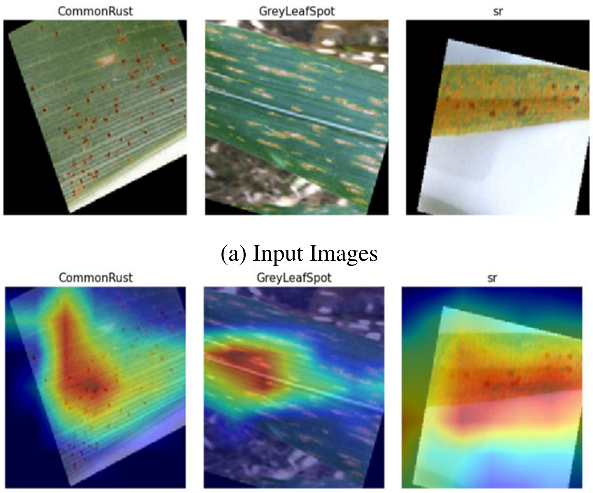
$$Weighted\ F1 - score = \sum_{i=1}^{N} W_i \times F1 - score_i \tag{8}$$

$$where,\ W_i = \frac{No\ of\ samples\ in\ each\ class_i}{Total\ number\ of\ samples} \tag{9}$$

In a similar way, the weighted average scores for precision and recall can also be computed.

E. VISUALIZATION OF ACTIVATIONS

In some cases, the models during training are trained on some background or irrelevant features, which gives a satisfactory model accuracy. Such situations can be checked by generating activation maps. The activation maps can help determine if the models focus on the right features. For example, the output of the visualization of activations created by Grad-cam [74] for the MobileNet model given three input images from the maize leaf dataset is as shown in Fig.10. The figure shows that the model focuses on the suitable feature for all the



(a) Input Images

(b) Activation maps generated from last convolutional layer

FIGURE 10. Activation maps generated by Grad-CAM given three input images (a) input images (b) activation maps generated from last convolutional layer.

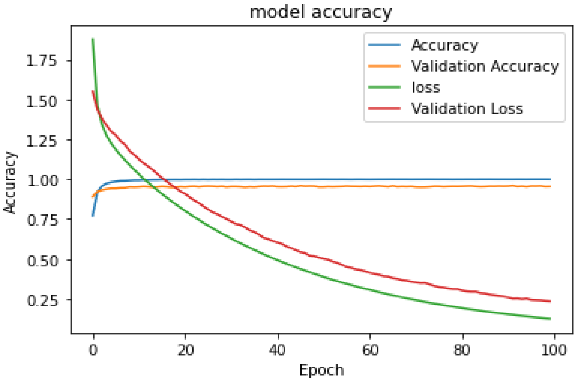


FIGURE 11. Accuracy and loss performance of the model xception on the maize leaf dataset.

input images from the maize leaf dataset, the images from three different classes CommonRust, GreyLeafSpot, and sr were given as input to the trained MobileNet model.

F. MAIZE

The performance of different models on the maize dataset is shown in Table 4. The maize dataset includes five classes, including the healthy class, for analysis. Xception, MobileNet, MobileNetV2, Inception V3, and InceptionResNetV2, models trained on the maize leaf dataset, showed better performance when compared to the other fine-tuned models. Xception showed a validation accuracy of 0.9608 and a testing accuracy of 0.9580. The InceptionV3 model showed a validation accuracy of 0.9468 and a testing accuracy of 0.9448. MobileNet showed a validation accuracy of 0.9484 and a testing accuracy of 0.9464. The MobileNetV2 model showed a validation accuracy of 0.9412 and a testing accuracy of 0.9404. The InceptionResNetV2 model showed a validation accuracy of 0.9336 and a testing accuracy of 0.9312. The fine-tuned Xception model performed best with a testing accuracy of 0.9580. The graphs showing the accuracy and loss performance of the models Xception and MobileNet showing better testing accuracy when compared to other fine-tuned models on the maize leaf dataset are shown in Fig. 11 and Fig. 12. The normalized confusion matrix of the models Xception and MobileNet on the maize leaf dataset is shown in Fig. 13 and Fig. 14. The performance of the models can be evaluated from the normalized confusion matrix by observing the diagonal elements. By referring to the color bar, the darker the color blue of the diagonal elements, the better the performance of the model.

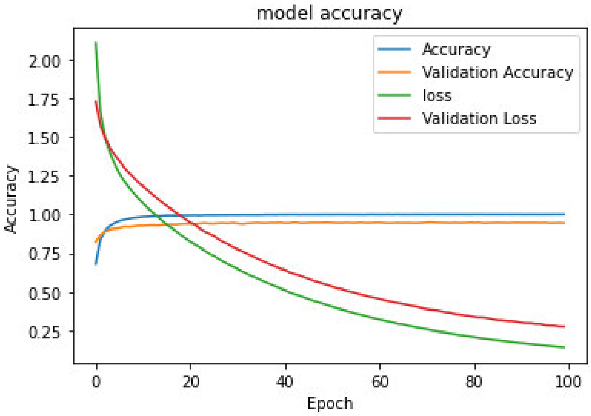


FIGURE 12. Accuracy and loss performance of the model MobileNet on the maize leaf dataset.

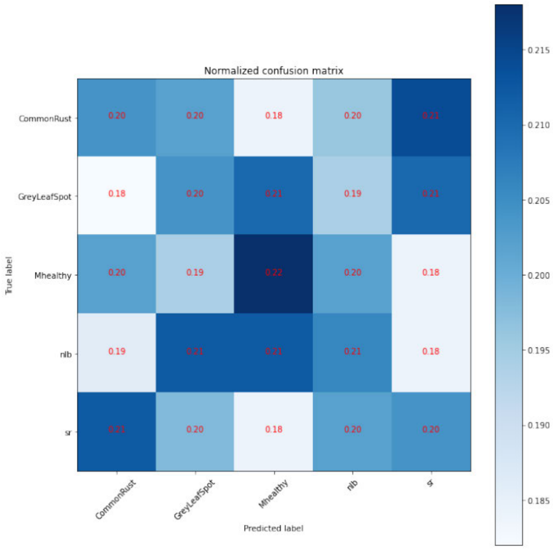


FIGURE 13. Normalized confusion matrix of the model xception on the maize leaf dataset.

In Fig. 13, for the Xception model on the maize leaf dataset, when observing the diagonal values, the Mhealthy gives a better value when compared to all the other classes. Confusion among classes can also be observed. For example,

TABLE 4. Application of state-of-the-art models on the maize leaf dataset.

Fine-tuned CNN Models	Training Accuracy	Validation Accuracy	Testing Accuracy	Optimizer	Learning Rate	No of Epochs
ResNet50	0.5709	0.5748	0.5604	SGD	0.001	100
ResNet 101	0.5939	0.5532	0.5180	SGD	0.001	100
MobileNetV2	0.9995	0.9412	0.9404	SGD	0.001	100
MobileNet	0.9991	0.9484	0.9464	SGD	0.001	100
Inception V3	0.9995	0.9468	0.9448	SGD	0.001	100
InceptionResNetV2	0.9994	0.9336	0.9312	SGD	0.001	100
Xception	0.9995	0.9608	0.9580	SGD	0.001	100
VGG16	0.9922	0.8976	0.8828	SGD	0.001	100
MRW-CNN(Proposed Model)	0.9518	0.9656	0.9704	SGD	0.001	100

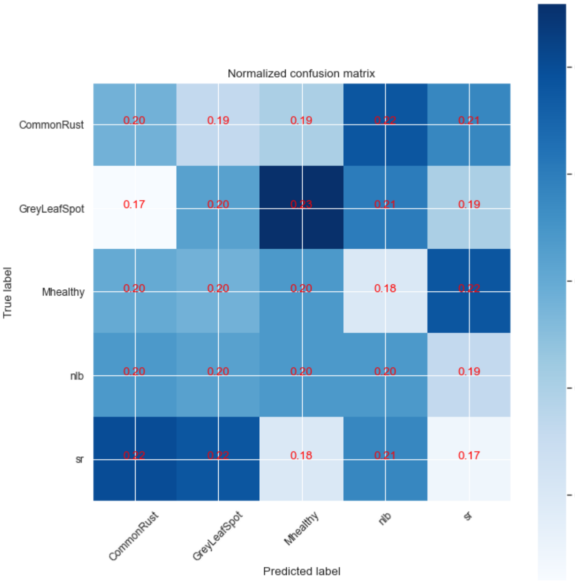


FIGURE 14. Normalized confusion matrix of the model MobileNet on the maize leaf dataset.

TABLE 5. Classification report of xception model on the maize leaf dataset.

	precision	recall	f1-score	support
CommonRust	0.21	0.21	0.21	500
GreyLeafSpot	0.20	0.21	0.20	500
Mhealthy	0.20	0.20	0.20	500
nlb	0.19	0.19	0.19	500
sr	0.19	0.19	0.19	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

TABLE 6. Classification report of mobilenet model on the maize leaf dataset.

	precision	recall	f1-score	support
CommonRust	0.22	0.22	0.22	500
GreyLeafSpot	0.19	0.20	0.20	500
Mhealthy	0.20	0.20	0.20	500
nlb	0.21	0.22	0.22	500
sr	0.19	0.18	0.18	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

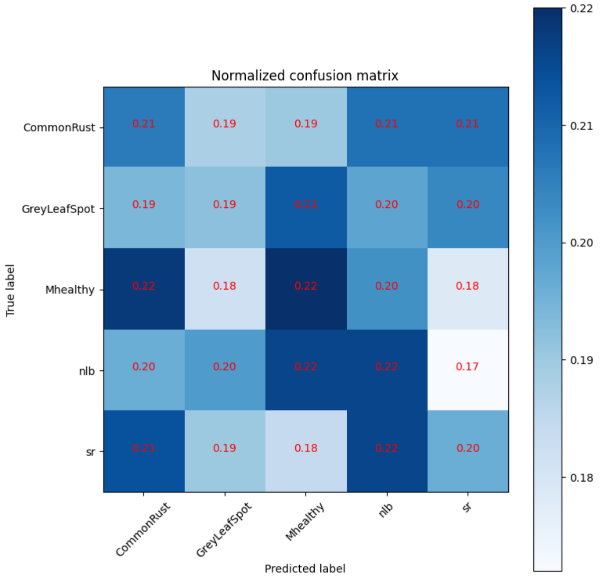


FIGURE 15. Normalized confusion matrix of the model MRW-CNN on the maize leaf dataset.

CommonRust and sr, nlb and GreyLeafSpot, nlb, Mhealthy, etc. In Fig. 14 for the MobileNet model on the maize leaf dataset, it can be observed that the diagonal values for all classes are nearly the same except for one. Therefore, it can be assumed that all classes perform equally the same except for the disease SouthernRust (sr). Some confusion among classes can also be observed; for example, GreyLeafSpot is highly confused with the class Mhealthy. There are also confusions between the classes sr and CommonRust and also sr and GreyLeafSpot, etc.

The proposed MRW-CNN model trained from scratch on the maize leaf dataset performed well; it showed a validation

TABLE 7. Classification report of MRW-CNN model on the maize leaf dataset.

	precision	recall	f1-score	support
CommonRust	0.21	0.21	0.20	500
GreyLeafSpot	0.21	0.20	0.21	500
Mhealthy	0.20	0.20	0.20	500
nlb	0.19	0.20	0.19	500
sr	0.19	0.19	0.19	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

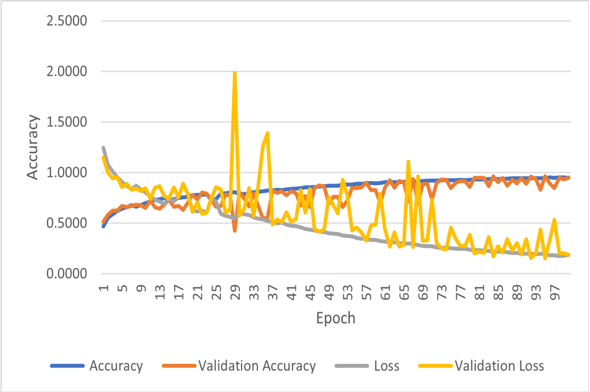


FIGURE 16. Accuracy and loss performance of the model MRW-CNN on the maize leaf dataset.

accuracy of 0.9656 and a 2% increase in testing accuracy of 0.9704 compared to the fine-tuned models. The normalized confusion matrix of the MRW-CNN model on the maize leaf dataset is shown in Fig. 15. In Fig. 15, when observing the diagonal elements, the Mhealthy and nlb classes of maize performed equally well when compared to the other classes. Confusion among classes can also be observed; for example, the class nlb is highly confused with Mhealthy, the Mhealthy class is confused with CommonRust, etc. The accuracy and loss performance of the MRW-CNN model on the maize leaf dataset is shown in Fig. 16.

The classification report of the Xception, MobileNet and the proposed MRW-CNN model on the maize leaf dataset is shown in Table 5, Table 6 and Table 7. The tables show the precision, recall, F1-score, and support scores for each class in the maize leaf dataset on the test data. The macro avg, micro average, and accuracy are also calculated. More details regarding how these values can be calculated in mentioned in Subsection D of Section VI.

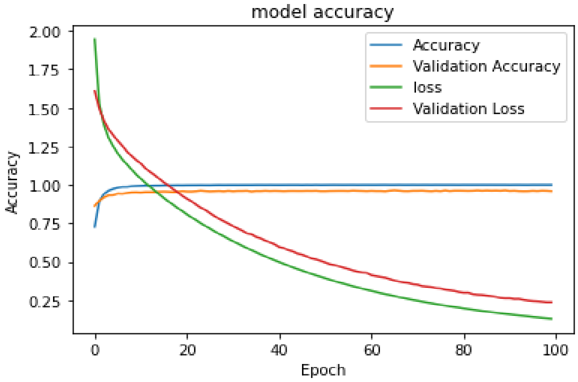


FIGURE 17. Accuracy and loss performance of the model inception V3 on the rice leaf dataset.

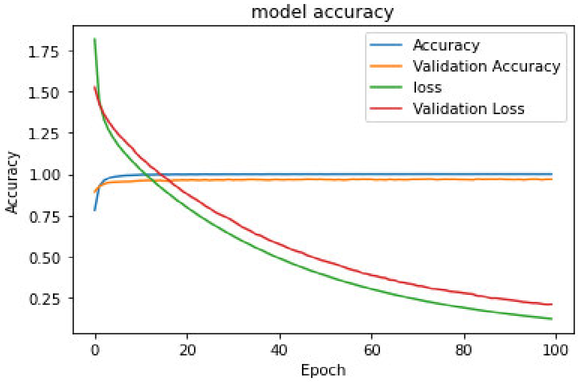


FIGURE 18. Accuracy and loss performance of the model xception on the rice leaf dataset.

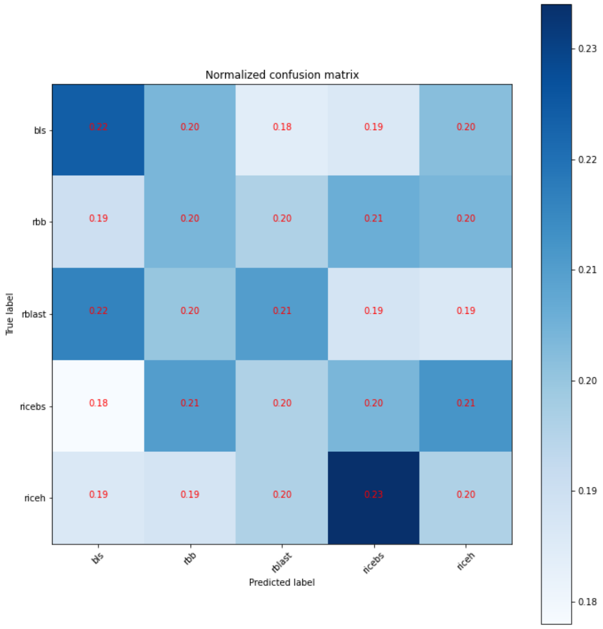


FIGURE 19. Normalized confusion matrix of the model inception V3 on the rice leaf dataset.

G. RICE

The performance of the state-of-the-art deep learning models on the rice leaf dataset is shown in Table 8. Out of all eight fine-tuned models trained on the rice leaf dataset, the models Xception, MobileNet, Inception V3, and MobileNetV2 showed validation and testing accuracy above 95% when compared to all the other models trained. The Xception model performed best when compared to all the other models.

TABLE 8. Application of the state-of-the-art models on the rice leaf dataset.

Fine-tuned CNN Models	Training Accuracy	Validation Accuracy	Testing Accuracy	Optimizer	Learning Rate	No of Epochs
ResNet 50	0.6657	0.6184	0.6232	SGD	0.001	100
ResNet 101	0.7075	0.6188	0.6068	SGD	0.001	100
MobileNetV2	0.9991	0.9624	0.9584	SGD	0.001	100
MobileNet	0.9994	0.9676	0.9572	SGD	0.001	100
InceptionV3	0.9998	0.9628	0.9620	SGD	0.001	100
InceptionResNetV2	0.9996	0.9492	0.9532	SGD	0.001	100
VGG16	0.9941	0.9148	0.9120	SGD	0.001	100
Xception	0.9999	0.9708	0.9728	SGD	0.001	100
MRW-CNN(Proposed Model)	0.9504	0.9729	0.9706	SGD	0.001	100

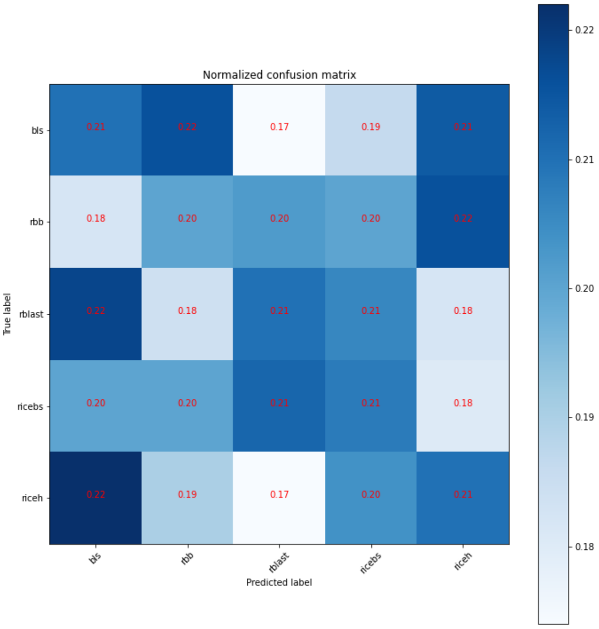


FIGURE 20. Normalized confusion matrix of the model xception on the rice leaf dataset.

TABLE 9. Classification report of xception model on the rice leaf dataset.

	precision	recall	f1-score	support
bls	0.21	0.22	0.22	500
rbb	0.21	0.20	0.21	500
rblast	0.19	0.18	0.19	500
ricebs	0.19	0.19	0.19	500
riceh	0.20	0.20	0.20	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

The MobileNetV2 model showed a validation accuracy of 0.9624 and a testing accuracy of 0.9584. The MobileNet model showed a validation accuracy of 0.9676 and a testing accuracy of 0.9572.

The Inception V3 model showed a validation accuracy of 0.9628 and a testing accuracy of 0.9620. The InceptionResNetV2 showed a validation accuracy of 0.9492 and a testing accuracy of 0.9532. The VGG16 model showed a validation accuracy of 0.9148 and a testing accuracy of 0.9120. The Xception model showed a validation accuracy of 0.9708 and a testing accuracy of 0.9728. The graphs

TABLE 10. Classification report of inception v3 model on the rice leaf dataset.

	precision	recall	f1-score	support
bls	0.20	0.21	0.20	500
rbb	0.20	0.20	0.20	500
rblast	0.19	0.19	0.19	500
ricebs	0.19	0.19	0.19	500
riceh	0.20	0.20	0.20	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

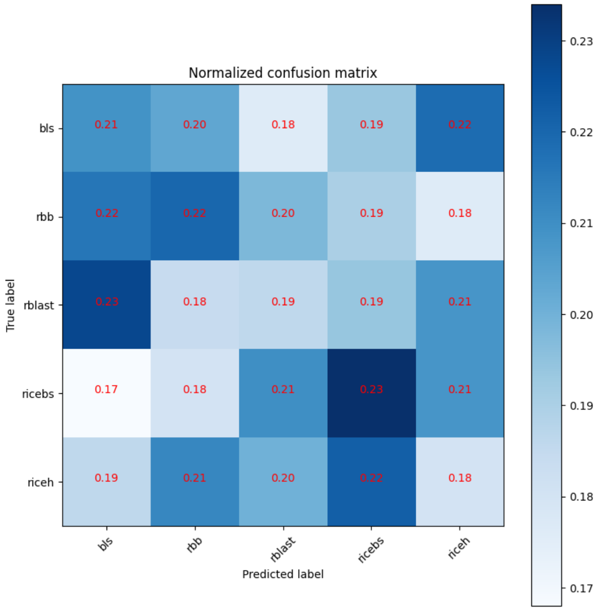


FIGURE 21. Normalized confusion matrix of the model MRW-CNN on the rice leaf dataset.

TABLE 11. Classification Report of MRW-CNN model on the rice leaf dataset.

	precision	recall	f1-score	support
bls	0.22	0.21	0.21	500
rbb	0.20	0.20	0.20	500
rblast	0.20	0.19	0.20	500
ricebs	0.22	0.23	0.23	500
riceh	0.20	0.20	0.20	500
accuracy			0.21	2500
macro avg	0.21	0.21	0.21	2500
weighted avg	0.21	0.21	0.21	2500

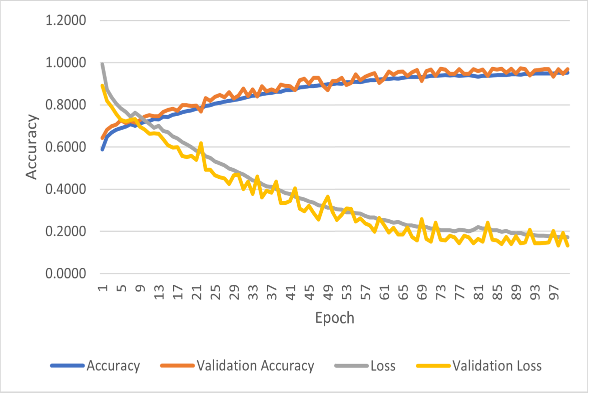


FIGURE 22. Accuracy and loss performance of the model MRW-CNN on the rice leaf dataset.

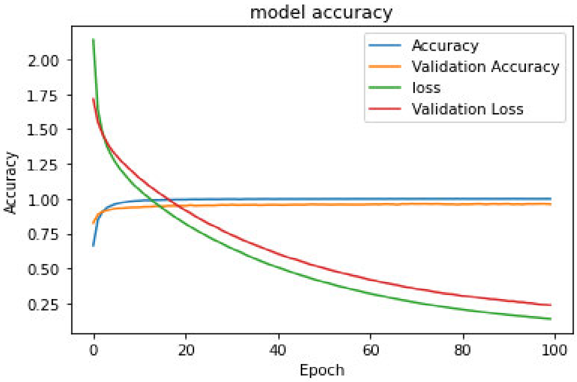


FIGURE 23. Accuracy and loss performance of the model mobilenet on the wheat leaf dataset.

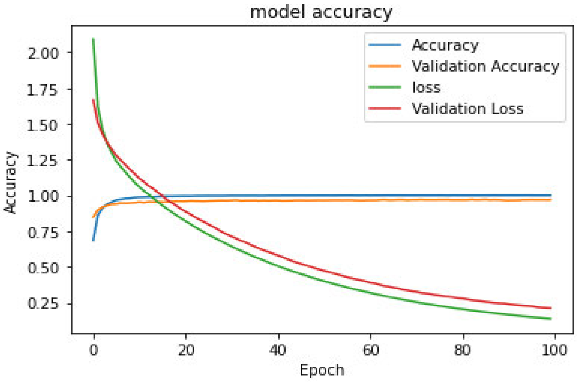


FIGURE 24. Accuracy and loss performance of the model mobilenetv2 on the wheat leaf dataset.

showing the accuracy and loss performance of the fine-tuned models trained on the rice leaf dataset with a testing accuracy of above 96% are shown in Fig. 17 and Fig. 18. The normalized confusion matrix of the fine-tuned models that showed a testing accuracy of above 96% on the rice leaf dataset is shown in Fig. 19 and Fig. 20.

Fig. 19 shows the normalized confusion matrix of the model Inception V3 on the rice leaf dataset. It can be observed from the confusion matrix that the class bls performs slightly better when compared to the other classes. Some classes are confused with each other, like rblast and bls, riceh and ricebs, etc. Similarly, Fig. 20 shows the normalized confusion matrix for the Xception model on the rice leaf dataset. It can be observed from the figure that all the classes perform equally the same except for rbb. Some classes that are highly confused with each other are rblast and bls, riceh, and bls, etc. The proposed MRW-CNN model trained from scratch performed equally well on the rice leaf dataset with a validation accuracy of 0.9729 and a testing accuracy of 0.9706 when compared with the fine-tuned models. The normalized confusion matrix of the MRW-CNN model on the rice leaf dataset is shown in Fig. 21. In Fig. 21 from the normalized confusion matrix, it can be observed that class ricebs performs slightly better when compared to the other classes. Confusion among classes can also be observed, for example, rblast and bls, riceh and ricebs, etc. The accuracy and loss performance of the MRW model on the rice leaf dataset is shown in Fig. 22. The classification report of the Xception, Inception V3, and the MRW-CNN model on the rice leaf dataset is shown in Table 9, Table 10, and Table 11. The classification report shows the precision, recall, F1-Score, support scores for all the classes in the rice leaf dataset for the test images. It also calculates the macro average, accuracy, and weighted average.

H. WHEAT

The performance of the state-of-the-art deep learning models on the wheat leaf dataset is shown in Table 12. Out of all eight models trained on the wheat leaf dataset, the models MobileNetV2, MobileNet, Inception V3, and InceptionResNetV2 showed a validation accuracy above 90% when compared to all the other models trained. The MobileNetV2 model showed a validation accuracy of 0.9724 and a testing accuracy of 0.9632. The MobileNet model showed a validation accuracy of 0.9632 and a testing accuracy of 0.9628. The Inception V3 model showed a validation accuracy of 0.9496 and a testing accuracy of 0.9508. The InceptionResNetV2 model showed a validation

TABLE 12. Application of the state-of-the-art models on the wheat leaf dataset.

Fine-tuned CNN Models	Training Accuracy	Validation Accuracy	Testing Accuracy	Optimizer	Learning Rate	No of Epochs
ResNet50	0.6028	0.5492	0.5536	SGD	0.001	100
ResNet 101	0.6382	0.5320	0.5299	SGD	0.001	100
MobileNetV2	0.9992	0.9724	0.9632	SGD	0.001	100
MobileNet	0.9995	0.9632	0.9628	SGD	0.001	100
InceptionV3	0.9996	0.9496	0.9508	SGD	0.001	100
InceptionResNetV2	0.9995	0.9516	0.9488	SGD	0.001	100
VGG16	0.9937	0.8972	0.8884	SGD	0.001	100
Xception	0.7377	0.4184	0.4018	SGD	0.001	100
MRW-CNN(Proposed Model)	0.9733	0.9793	0.9808	SGD	0.001	100

TABLE 13. Classification report of mobilenet model on the wheat leaf dataset.

	precision	recall	f1-score	support
leafrust	0.21	0.21	0.21	500
powderymildew	0.21	0.22	0.22	500
striperust	0.20	0.20	0.20	500
tanspot	0.20	0.20	0.20	500
wh	0.20	0.20	0.20	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

TABLE 14. Classification report of MobileNetV2 model on the wheat leaf dataset.

	precision	recall	f1-score	support
leafrust	0.20	0.20	0.20	500
powderymildew	0.21	0.22	0.21	500
striperust	0.20	0.20	0.20	500
tanspot	0.20	0.20	0.20	500
wh	0.20	0.20	0.20	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

TABLE 15. Classification report of MRW-CNN model on the wheat leaf dataset.

	precision	recall	f1-score	support
leafrust	0.21	0.22	0.21	500
powderymildew	0.21	0.20	0.21	500
striperust	0.20	0.20	0.20	500
tanspot	0.19	0.19	0.19	500
wh	0.20	0.20	0.20	500
accuracy			0.20	2500
macro avg	0.20	0.20	0.20	2500
weighted avg	0.20	0.20	0.20	2500

accuracy of 0.9516 and a testing accuracy of 0.9488. The graphs showing the accuracy and loss performance of the fine-tune models trained on the wheat leaf dataset with a testing accuracy of above 96% are shown in Fig. 23 and Fig. 24.

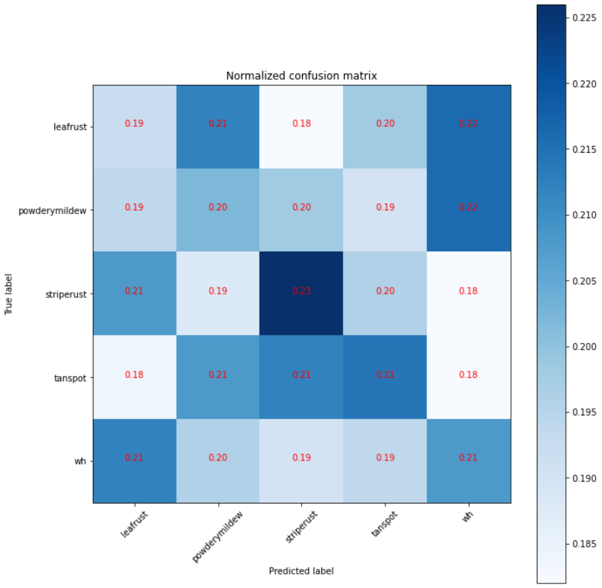


FIGURE 25. Normalized confusion matrix of the model mobilenet on the wheat leaf dataset.

The normalized confusion matrix of the fine-tuned models that showed a testing accuracy of above 96% on the wheat leaf dataset are shown in Fig. 25 and Fig. 26. Fig. 25 shows the normalized confusion matrix of the MobileNet model on the wheat leaf dataset. From the confusion matrix, it can be observed that the striperust class performed best when compared to all the other classes in the test data. Confusions among classes also were observed, for example, leafrust and wh, powderymildew and wh, etc. Fig. 26 shows the normalized confusion matrix of the MobileNetV2 model on the wheat leaf dataset.

The stripe rust class performed well when compared to others. Some confusing classes were wh and leafrust, powderymildew and wh, leafrust and wh, etc. The proposed MRW-CNN model trained from scratch on the wheat leaf dataset performed well; it showed a validation accuracy of 0.9793 and a 2% increase in testing accuracy of 0.9808 when compared to the fine-tuned models. The normalized confusion matrix of the MRW-CNN model on the wheat leaf

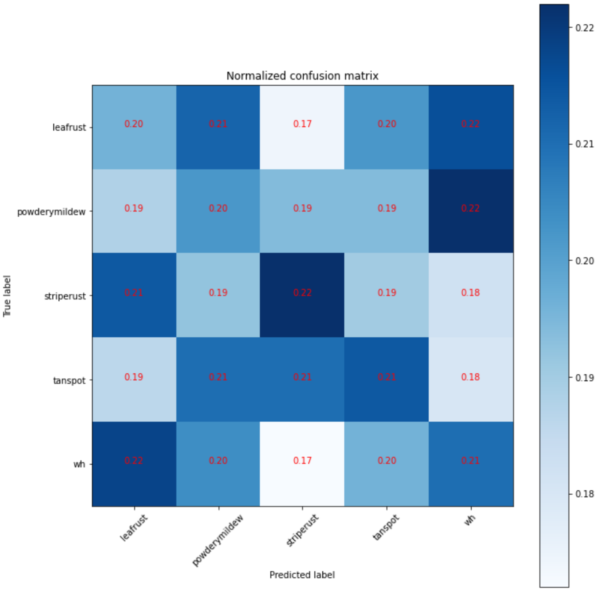


FIGURE 26. Normalized confusion matrix of the model mobilenetv2 on the wheat leaf dataset.

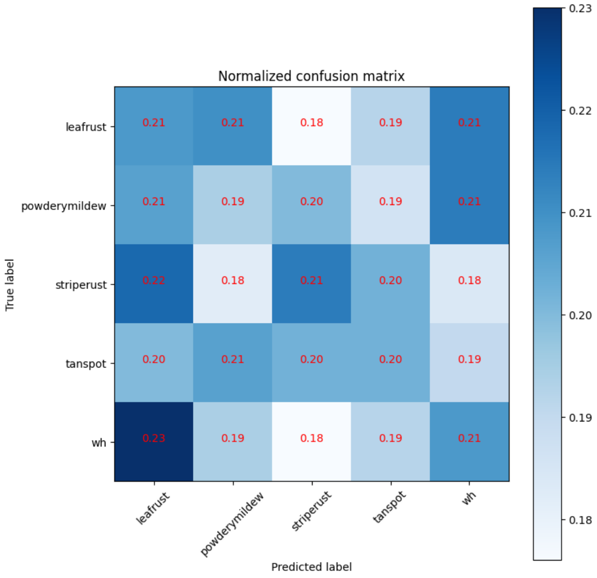


FIGURE 27. Normalized confusion matrix of the model MRW-CNN on the wheat leaf dataset.

dataset is shown in Fig. 27. In Fig. 27, when observing the diagonal elements, the leafrust, striperust, and wh class of wheat performed equally well when compared to the other classes. Confusion among classes can also be observed; for example, the class wh is highly confused with leafrust, the leafrust class is confused with wh, etc. The accuracy and loss performance of the MRW-CNN model on the wheat leaf dataset is shown in Fig. 28. The classification report of the MobileNet, MobileNetV2, and the MRW-CNN model on the wheat leaf dataset is shown in Table 13, Table 14 and Table 15. The tables show the precision, recall, F1-score and support scores for each class in the wheat leaf dataset on the test data. The macro avg, micro average, and accuracy are also calculated.

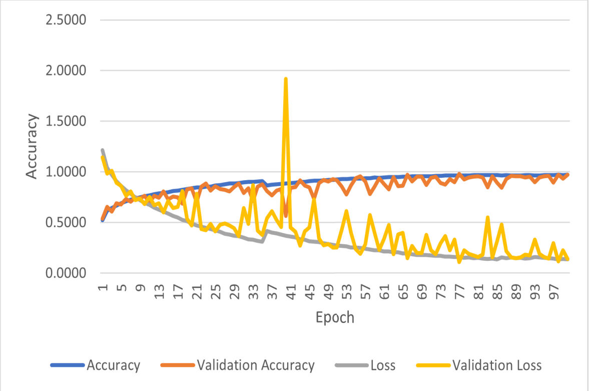


FIGURE 28. Accuracy and loss performance of the model MRW-CNN on the wheat leaf dataset.

VII. CASE STUDY OF THE APPLICATION OF THE PROPOSED METHOD

Computer vision technology will be a need in autonomous futuristic farms. CNN based computer vision tasks have shown great performance in recent years. Therefore, there is an increased need to concentrate on the practical applications of the system developed. If deployed properly, the proposed method can help reduce crop loss and increase yield. A plant protection system APP based on food grains can be developed using the proposed system and the datasets developed. It can be useful for detecting the diseases affecting the food grains of rice, wheat, and maize.

Depending on the disease detected, appropriate curable measures can be suggested. Identifying the disease at the correct stage can help the users decide on the amount of pesticide to be used and in what amount. The APP can be trained to identify diseases with similar symptoms by asking users additional questions about the weather conditions, the location of the disease on the leaf, etc. By knowing more about the weather conditions and the location of the disease. It becomes easy to distinguish between diseases showing similar symptoms to some extent.

Additional information regarding the weather conditions that can favor the spread or advancement of the disease can be given to the users. Suggestions on the pesticides that can be used and the amount can also be made available. Small-scale growers are the people who benefit more from these kinds of solutions available for diagnosing plant diseases when they are not able to get help from experts all the time. The overall architecture of the plant protection system that can be developed is shown in Fig. 29.

Fig. 29 discusses the possible use cases the user can give the plant disease protection system. In case 1, the user provides a diseased leaf to the system to detect the disease affecting the leaf along with its stage. After the detection

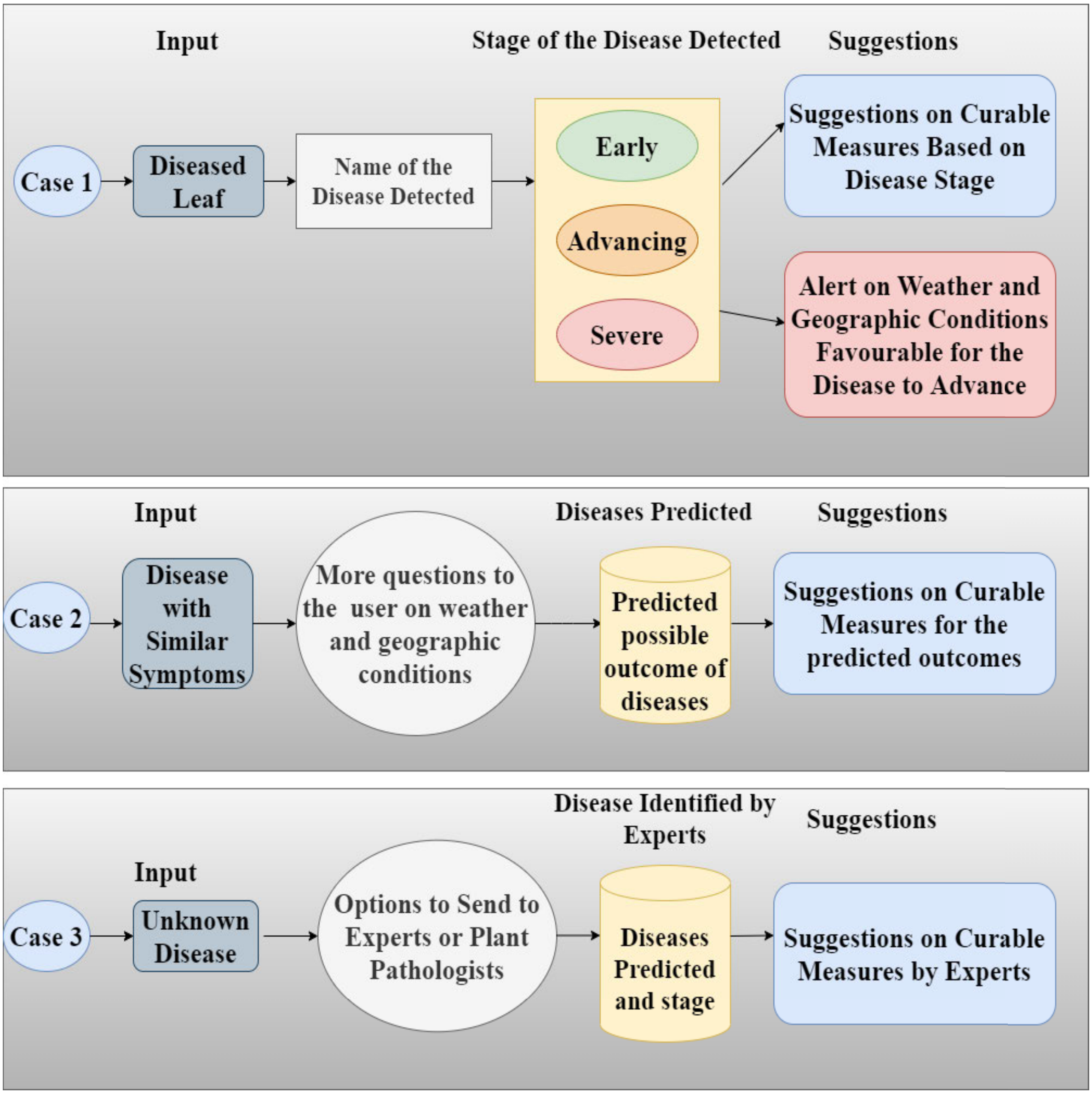

Input
Stage of the Disease Detected
Suggestions
Case 1
Diseased Leaf
Name of the Disease Detected
Early
Advancing
Severe
Suggestions on Curable Measures Based on Disease Stage
Alert on Weather and Geographic Conditions Favourable for the Disease to Advance
Input
Diseases Predicted
Suggestions
Case 2
Disease with Similar Symptoms
More questions to the user on weather and geographic conditions
Predicted possible outcome of diseases
Suggestions on Curable Measures for the predicted outcomes
Disease Identified by Experts
Suggestions
Input
Case 3
Unknown Disease
Options to Send to Experts or Plant Pathologists
Diseases Predicted and stage
Suggestions on Curable Measures by Experts

FIGURE 29. Application of the proposed system.

of the disease and its stage. The system also provides suggestions regarding the curable measures and alerts on the weather conditions that can be favorable for advancing the disease. In case 2, when a diseased leaf has similar symptoms of one or two diseases, more questions can be asked to the users about the weather and geographical conditions. Based on this information, possible predictions of diseases can be made by the system. Also, suggestions for curable measures for these predicted outcomes can be made.

In case 3, when the system encounters an unknown disease, options can be given to the user to send it to an expert or plant pathologist for analysis. The system learns from the disease identified by the experts and their suggestions. The system learns from the communication between the users or growers and the experts.

VIII. CONCLUSION AND FUTURE WORK

A. CONCLUSION

In the current work, real-time datasets for the food grains, rice, wheat, and maize were formed by considering images from different datasets and internet sources. Importance was given to real-life images of different disease severity levels, which is important for farmers or growers to make appropriate decisions at the early stages of infection. Identification of disease at the early stage of infection can reduce yield loss and financial loss.

According to experimental results, CNN-based deep learning models pre-trained on the ImageNet dataset, with deep transfer learning strategy, downloaded from Keras and fine-tuned, and the proposed MRW-CNN model performs well in identifying and classifying rice, maize, and wheat diseases considered in the current work. All the models were trained for 100 epochs.

During the development of the dataset, real-life images were concentrated on more, rather than processed images. Selected images from the datasets were pre-processed to remove foreign objects if present. The number of images collected under each class was 100 which was not enough for training a deep learning model. So, the technique of data augmentation was applied to all images per class to increase the size of the dataset. The original images were combined with the augmented images, and then the dataset formed was splitted 80% for training and 10% each for validation and testing for all three food grains: rice, wheat, and maize. The datasets were then applied to different fine-tuned CNN models and the proposed MRW-CNN model. The models were evaluated using performance metrics like accuracy, f1-score, precision, recall, etc.

B. FUTURE WORK

It can be observed that the datasets we developed have some limitations. Although most of the images in the dataset are real-life, some are pre-processed. In real-life scenarios, more factors will influence the results than those concentrated on developing the datasets in the current work. Therefore, as future work, more importance can be given to factors like how the impact of climate change can affect plant diseases collection of leaf diseased images during different climatic conditions.

Some of the other future works can be annotated images from the dataset can be used in object detection models like MaskRCNN to detect disease severity levels. A CNN based deep learning system can be developed to detect and classify diseases in food grains; the dataset can be further extended by adding images of more food grains, images of multiple diseases on the same leaf, etc.

REFERENCES

[1] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Eng.*, vol. 144, pp. 52–60, Apr. 2016.

[2] D. S. Joseph, P. M. Pawar, and R. Pramanik, "Intelligent plant disease diagnosis using convolutional neural network: A review," *Multimedia Tools Appl.*, vol. 82, no. 14, pp. 21415–21481, Jun. 2023.

[3] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105393.

[4] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–8, Jul. 2017.

[5] O. Getch. *Rice Knowledge Bank: Your Information Source for Rice Farming*. Accessed: Jan. 30, 2023. [Online]. Available: http://www.knowledgebank.irri.org/step-by-step-production/growth/pests-and-diseases/diseases

[6] E. D. Wolf and J. P. Shroyer. *Wheat Disease Identification*. Accessed: Nov. 25, 2022. [Online]. Available: https://www.bookstore.ksre.ksu.edu/pubs/MF2994.pdf

[7] S. Mahadik, P. M. Pawar, and R. Muthalagu, "Efficient intelligent intrusion detection system for heterogeneous Internet of Things (HetIoT)," *J. Netw. Syst. Manag.*, vol. 31, no. 1, p. 2, Jan. 2023.

[8] K. Syama, J. A. A. Jothi, and N. Khanna, "Automatic disease prediction from human gut metagenomic data using boosting GraphSAGE," *BMC Bioinf.*, vol. 24, no. 1, p. 126, Mar. 2023.

[9] H. Hussain and P. S. Tamizharasan, "The impact of cascaded optimizations in CNN models and end-device deployment," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, J. Gummeson, S. I. Lee, J. Gao, and G. Xing, Eds. Boston, MA, USA, 2022, pp. 954–961.

[10] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019.

[11] Z. Libo, H. Tian, G. Chunyun, and M. Elhoseny, "Real-time detection of Cole diseases and insect pests in wireless sensor networks," *J. Intell. Fuzzy Syst.*, vol. 37, no. 3, pp. 3513–3524, Oct. 2019.

[12] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[13] M. Shoaib, B. Shah, S. Ei-Sappagh, A. Ali, A. Ullah, F. Alenezi, T. Gechev, T. Hussain, and F. Ali, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers Plant Sci.*, vol. 14, Mar. 2023, Art. no. 1158933.

[14] M. Shoaib, B. Shah, T. Hussain, A. Ali, A. Ullah, F. Alenezi, T. Gechev, F. Ali, and I. Syed, "A deep learning-based model for plant lesion segmentation, subtype identification, and survival probability estimation," *Frontiers Plant Sci.*, vol. 13, Dec. 2022, Art. no. 1095547.

[15] M. Akbar, M. Ullah, B. Shah, R. U. Khan, T. Hussain, F. Ali, F. Alenezi, I. Syed, and K. S. Kwak, "An effective deep learning approach for the classification of bacteriosis in peach leave," *Frontiers Plant Sci.*, vol. 13, p. 4723, Nov. 2022.

[16] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," 2015, *arXiv:1511.08060*.

[17] Q. Zeng, X. Ma, B. Cheng, E. Zhou, and W. Pang, "GANs-based data augmentation for citrus disease severity detection using deep learning," *IEEE Access*, vol. 8, pp. 172882–172891, 2020.

[18] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Comput. Electron. Agricult.*, vol. 154, pp. 18–24, Nov. 2018.

[19] Q. Wu, Y. Chen, and J. Meng, "DCGAN-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 98716–98728, 2020.

[20] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agricult.*, vol. 161, pp. 272–279, Jun. 2019.

[21] R. Kumar, U. Mina, R. Gogoi, A. Bhatia, and R. C. Harit, "Effect of elevated temperature and carbon dioxide levels on maydis leaf blight disease tolerance attributes in maize," *Agricult., Ecosystems Environ.*, vol. 231, pp. 98–104, Sep. 2016.

[22] S. Nigam, R. Jain, S. Marwaha, A. Arora, M. A. Haque, A. Dheeraj, and V. K. Singh, "Deep transfer learning model for disease identification in wheat crop," *Ecological Informat.*, vol. 75, Jul. 2023, Art. no. 102068.

[23] M. K. Vishal, D. Tamboli, A. Patil, R. Saluja, B. Banerjee, A. Sethi, D. Raju, S. Kumar, R. N. Sahoo, V. Chinnusamy, and J. Adinarayana, "Image-based phenotyping of diverse rice (*Oryza sativa* L.) genotypes," 2020, *arXiv:2004.02498*.

[24] A. Chug, A. Bhatia, A. P. Singh, and D. Singh, "A novel framework for image-based plant disease detection using hybrid deep learning approach," *Soft Comput.*, vol. 27, no. 18, pp. 13613–13638, Sep. 2023.

[25] M. A. Haque, S. Marwaha, C. K. Deb, S. Nigam, A. Arora, K. S. Hooda, P. L. Soujanya, S. K. Aggarwal, B. Lall, M. Kumar, S. Islam, M. Panwar, P. Kumar, and R. C. Agrawal, "Deep learning-based approach for identification of diseases of maize crop," *Sci. Rep.*, vol. 12, no. 1, p. 6334, Apr. 2022.

[26] S. Das, A. Biswas, and P. Sinha, "Deep learning analysis of rice blast disease using remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[27] M. Shoaib, T. Hussain, B. Shah, I. Ullah, S. M. Shah, F. Ali, and S. H. Park, "Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease," *Frontiers Plant Sci.*, vol. 13, Oct. 2022, Art. no. 1031748.

[28] 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.

[29] H. Wu, T. Wiesner-Hanks, E. L. Stewart, C. DeChant, N. Kaczmar, M. A. Gore, R. J. Nelson, and H. Lipson, "Autonomous detection of plant disease symptoms directly from aerial imagery," *Plant Phenome J.*, vol. 2, no. 1, pp. 1–9, Jan. 2019.

[30] J. Sun, Y. Yang, X. He, and X. Wu, "Northern maize leaf blight detection under complex field environment based on deep learning," *IEEE Access*, vol. 8, pp. 33679–33688, 2020.

[31] S. Mishra, R. Sachan, and D. Rajpal, "Deep convolutional neural network based detection system for real-time corn plant disease recognition," *Proc. Comput. Sci.*, vol. 167, pp. 2003–2010, Jan. 2020.

[32] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.

[33] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Comput. Electron. Agricult.*, vol. 142, pp. 369–379, Nov. 2017.

[34] R. Qiu, C. Yang, A. Moghimi, M. Zhang, B. J. Steffenson, and C. D. Hirsch, "Detection of fusarium head blight in wheat using a deep neural network and color imaging," *Remote Sens.*, vol. 11, no. 22, p. 2658, Nov. 2019.

[35] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017.

[36] F. Jiang, Y. Lu, Y. Chen, D. Cai, and G. Li, "Image recognition of four rice leaf diseases based on deep learning and support vector machine," *Comput. Electron. Agricult.*, vol. 179, Dec. 2020, Art. no. 105824.

[37] D. P. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *Proc. 7th ACM IKDD CoDS 25th COMAD*, 2019, pp. 249–253.

[38] S. Riyaz. *Rice Leafs: An Image Collection Four Rice Diseases*. Accessed: Jan. 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/shayanriyaz/riceleafs

[39] O. Getch. *Wheat Leaf Dataset: Disease Affected and Healthy Wheat Leaf*. Accessed: Jan. 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/olyadgetch/wheat-leaf-dataset

[40] *Rice Knowledge Bank*. Accessed: Oct. 12, 2022. [Online]. Available: http://www.knowledgebank.irri.org/decision-tools/rice-doctor/rice-doctor-fact-sheets

[41] D. Groth. *Brown Spot of Rice*. Accessed: Oct. 6, 2022. [Online]. Available: https://www.forestryimages.org/browse/detail.cfm?imgnum=5410673

[42] *Leaf Rust*. Accessed: Oct. 12, 2022. [Online]. Available: https://agritech.tnau.ac.in/crop_protection/wheat/crop_prot_crop

[43] *Leaf Rust*. Accessed: Oct. 12, 2022. [Online]. Available: https://www.cropscience.bayer.co.nz/pests/diseases/leaf-rust—wheat

[44] *Leaf Rust of Wheat*. Accessed: Oct. 12, 2022. [Online]. Available: https://extensionaus.com.au/FieldCropDiseasesVic/docs/identification-management-of-field-crop-diseases-in-victoria/foliar-diseases-of-wheat/leaf-rust-of-wheat/

[45] *Powdery Mildew*. Accessed: Oct. 12, 2022. [Online]. Available: https://www.cropscience.bayer.co.nz/pests/diseases/powdery-mildew—wheat

[46] *New Permits Aid Control of Powdery Mildew in Wheat*. Accessed: Oct. 12, 2022. [Online]. Available: https://www.graincentral.com/cropping/new-permits-aid-control-of-powdery-mildew-in-wheat/

[47] *Scout Wheat Fields for Early Disease Detection*. Accessed: Oct. 12, 2022. [Online]. Available: https://cropwatch.unl.edu/2021/scout-wheat-fields-early-disease-detection

[48] *Powdery Mildew Symptoms*. Accessed: Oct. 12, 2022. [Online]. Available: https://doraagri.com/powdery-mildew-on-plants/

[49] *Stripe Rust (Yellow Rust) of Wheat*. Accessed: Oct. 12, 2022. [Online]. Available: https://extension.uga.edu/publications/detail.html?number=C960&

[50] *Tan Spot*. Accessed: Oct. 12, 2022. [Online]. Available: https://andersonscanada.com/2017/05/17/wheat-leaf-diseases-2/

[51] *Tan Spot*. Accessed: Oct. 12, 2022. [Online]. Available: https://extension.okstate.edu/programs/digital-diagnostics/plant-diseases/tan-spot.html

[52] *Tan Spot of Wheat*. Accessed: Oct. 12, 2022. [Online]. Available: https://extensionpublications.unl.edu/assets/html/g429/build/g429.htm

[53] *Tan Spot*. Accessed: Oct. 12, 2022. [Online]. Available: https://cropscience.bayer.co.uk/threats/diseases/cereal-diseases/tan-spot/

[54] *Common Rust of Corn*. Accessed: Oct. 7, 2022. [Online]. Available: https://ohioline.osu.edu/factsheet/plpath-cer-02

[55] *Common Corn Rust Puccinia Sorghi Schwein*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.forestryimages.org/browse/detail.cfm?imgnum=1538028

[56] A. Sisson. *Common Corn Rust Puccinia Sorghi Schwein*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.forestryimages.org/browse/detail.cfm?imgnum=5608271

[57] G. Holmes. *Common Corn Rust Puccinia Sorghi Schwein*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.forestryimages.org/browse/detail.cfm?imgnum=5606719

[58] A. Robertson. *Gray Leaf Spot*. Accessed: Oct. 7, 2022. [Online]. Available: https://crops.extension.iastate.edu/cropnews/2018/06/weather-conditions-brown-spot-and-node-rot-and-gray-leaf-spot

[59] *Northern Corn Leaf Blight, Grey Leaf Spot Top Ontario Corn Diseases*. Accessed: Nov. 22, 2022. [Online]. Available: https://www.topcropmanager.com/northerncornleafblightgreyleafspottop-ontariocorndiseases19935/

[60] *Managing Northern Corn Leaf Blight*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.cropscience.bayer.ca/articles/2021/northern-corn-leaf-blight

[61] Chroanch. *Northern Corn Leaf Blight*. Accessed: Oct. 7, 2022. [Online]. Available: https://commons.wikimedia.org/Northern_corn_leaf_blight.JPG

[62] D. Mueller. *Northern Corn Leaf Blight of Corn*. Accessed: Oct. 7, 2022. [Online]. Available: https://cropprotectionnetwork.org/encyclopedia/northern-corn-leaf-blight-of-corn

[63] A. Robertson. *Northern Corn Leaf Blight of Corn*. Accessed: Oct. 7, 2022. [Online]. Available: https://cropprotectionnetwork.org/encyclopedia/northern-corn-leaf-blight-of-corn

[64] *Southern Rust*. Accessed: Oct. 7, 2022. [Online]. Available: https://cropwatch.unl.edu/plantdisease/corn/southern-rust

[65] *Southern Rust of Corn*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.pioneer.com/us/agronomy/southern_rust_cropfocus.html

[66] *Mississippi Crop Situation*. Accessed: Oct. 7, 2022. [Online]. Available: https://www.mississippi-crops.com/2012/07/21/southern-corn-rust-is-a-late-fungicide-application-warranted/

[67] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "NIH image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7, pp. 671–675, Jul. 2012.

[68] *Imagej*. Accessed: Oct. 6, 2022. [Online]. Available: https://imagej.net/software/imagej/

[69] *Make Sense Ai Tool*. Accessed: Oct. 6, 2022. [Online]. Available: https://www.makesense.ai/

[70] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[71] I. Chelliah. *Confusion Matrix for Multiclass Classification*. Accessed: Feb. 21, 2023. [Online]. Available: https://medium.com/mlearning-ai/confusion-matrix-for-multiclass-classification-f25ed7173e66

[72] *Micro-Average & Macro-Average Scoring Metrics—Python*. Accessed: Feb. 21, 2023. [Online]. Available: https://vitalflux.com/micro-average-macro-average-scoring-metrics-multi-class-classification/

[73] *F1 Score in Machine Learning: Intro & Calculation*. Accessed: Feb. 21, 2023. [Online]. Available: https://www.v7labs.com/blog/f1-score-guide

[74] R. R. Selvaraju, P. Chattopadhyay, M. Elhoseiny, T. Sharma, D. Batra, D. Parikh, and S. Lee, "Choose your neuron: Incorporating domain knowledge through neuron-importance," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 526–541.

DIANA SUSAN JOSEPH (Student Member, IEEE) received the B.Tech. degree in computer science and engineering from the Rajagiri School of Engineering and Technology, in 2008, and the M.Tech. degree in computer and communication from Karunya University, Coimbatore, in 2011. She is currently pursuing the Ph.D. degree in computer science and engineering with the Birla Institute of Technology and Science, Dubai. She was an Assistant Professor with the Department of Computer Science and Engineering, in 2012. Her research interests include image processing, machine learning, and deep learning.

PRANAV M. PAWAR (Member, IEEE) received the Graduate degree in computer engineering from Dr. Babasaheb Ambedkar Technological University, Maharashtra, India, in 2005, the master's degree in computer engineering from Pune University, in 2007, and the Ph.D. degree in wireless communication from Aalborg University, Denmark, in 2016. His Ph.D. thesis received nomination for Best Thesis Award from Aalborg University. Currently, he is an Assistant Professor with the Department of Computer Science, Birla Institute of Technology and Science, Dubai, before to BITS, he was a Postdoctoral Fellow with Bar-Ilan University, Israel, from March 2019 to October 2020, in an area of wireless communication and deep leaning. He was recipient of Outstanding Postdoctoral Fellowship from Israel Planning and Budgeting Committee. He was an Associate Professor with MIT ADT University, Pune, from 2018 to 2019, and also as an Associate Professor with the Department of Information Technology, STES's Shrimati Kashibai Navale College of Engineering, Pune, from 2008 to 2018. From 2006 to 2007, he was a System Executive with POS-IPC, Pune, India. He received Recognition from Infosys Technologies Ltd., for contribution in Campus Connect Program and also received different funding for research and attending conferences at international level. He has published more than 40 papers at national and international level. He is also IBM DB2 and IBM RAD certified professional and completed NPTEL certification in different subjects. His research interests include energy efficient MAC for WSN, QoS in WSN, wireless security, green technology, computer architecture, database management systems, and bioinformatics.

KAUSTUBH CHAKRADEO received the Bachelor of Engineering degree in information technology from Savitribai Phule Pune University (formerly Pune University), in 2019, and the Master of Science degree in artificial intelligence from Radboud University, The Netherlands, in 2022. He is currently pursuing the Ph.D. degree with the Section of Epidemiology, Department of Public Health, University of Copenhagen. His research interests include computer vision, deep learning, AI ethics, and infectious disease modeling.