An Implementation and Analysis of Deep Learning Models for the Detection of Wheat Diseases

by

Ahmad Zubair
19101147
Sharmin Akter Keya
19301074
Tasnia Zarin Shailee
19101145
Syed Mahathir Md. Lenin
18301268
Dhruba Nandi
19301256

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

Declaration

It is hereby declared that

1. The report submitted is our own original work while completing degree at Brac University.

2. The report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Zubair

Ahmad Zubair
19101147

Keya

Sharmin Akter Keya
19301074

Shailee.

Tasnia Zarin Shailee
19101145

Syed Mahathir Md. Lenin
18301268

Dhruba

Dhruba Nandi
19301256

Approval

The thesis titled "An Implementation and Analysis of Deep Learning Models for the Detection of Wheat Diseases" submitted by

1. Ahmad Zubair (19101147)
2. Sharmin Akter Keya (19301074)
3. Tasnia Zarin Shailee (19101145)
4. Syed Mahathir Md. Lenin (18301268)
5. Dhruba Nandi (19301256)

Of Summer 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September, 2023.

Examining Committee:

Supervisor:
(Member)

Iqbal

Muhammad Iqbal Hossain, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi
Associate Professor and Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

Detecting a disease visually is a time-consuming and error-prone operation, and in the agricultural arena, for disease control, crop yield loss prediction, and global food security, automatic and accurate evaluation of disease severity in crops is a particularly demanding study area. Deep Learning (DL), the latest innovation in the era of Artificial Intelligence (AI), is promising for fine-grained categorization of crop diseases since it eliminates labor-intensive feature extraction and segmentation. To diagnose the disease from photos, multiple pretrained models which are ResNet50, EfficientNetB0 and InceptionV3 along with ViT and a hybrid CNN model have been trained on the wheat disease dataset. Again, an ensemble model of the hybrid CNN and the ViT has been proposed which has been compared with all the other models and the proposed model gets the highest accuracy of 99.34% among all the models.

Keywords: Deep Learning; Machine Learning; Wheat diseases; Prediction; Data augmentation; Vision Transformer; CNN; Ensemble

Acknowledgement

We like to show our deepest appreciation to the people who played critical roles in the preparation of this thesis. Firstly we want to express our heartfelt thanks to Muhammad Iqbal Hossain sir whose guidance and support were important in defining this research. His invaluable ideas, leadership and commitment to our academic development have been incredibly inspirational. We are also appreciative to our friends for their help along this adventure. Their collaborative spirit, brainstorming and encouragement have given this work a depth that would not have been possible otherwise. In addition, Abu Noman Faruq Ahmmed who provided the private data class(Wheat Blast) for this research. Finally, we want to show our heartfelt gratitude to our family for being an ongoing resource of emotional support and motivation for us. Their faith in us has been the driving factor behind our perseverance and determination even through the most difficult circumstances. This thesis would not have been possible without the joint efforts of these individuals for which we are grateful.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN : Convolutional Neural Network

$DCNN$: Deep Convolutional Neural Network

DL : Deep Learning

FHB : Fusarium head blight

$k-NN$: K-Nearest Neighbour

KNN : K-Nearest Neighbour

ML : Machine Learning

NB : Naive Bayes

NIR : Near-infrared

NN : Neural Network

$RCNN$: Region-Based Convolutional Neural Network

RF : Random Forest

RGB : Red-Green-Blue

SVM : Support Vector Machine

ViT : Vision Transformer

Chapter 1

Introduction

After Bangladesh got independence in 1971, rice was the main crop that people used to eat. During the period from 1971 to 1975, the rice price was increasing globally, which caused food shortages in Bangladesh. In addition, Bangladesh was already lagging behind in food production due to the War of Liberation along with natural disasters and population growth. It was then decided rice alone could not fulfill the nation's food demand.

As a result, wheat was chosen as an alternative winter crop. Since then wheat has been growing in popularity in Bangladesh, but climate change has a significant impact on Bangladesh and so the likelihood of growing wheat and potatoes would be substantially hampered by a shift in the average temperature of 2-4 °C, and output loss may surpass 60% of the possible yields [1].

Wheat consumption in 2021 has seen twice as much demand as in 2015 according to [32]. The reasons behind this are the good quality flour being cheaper than medium quality rice, the increase of the obese and diabetic patients, and the rise in the popularity of bakery products such as biscuits and cakes. Thus, the imports have seen a 116% rise in the past 6 years. However, the production of wheat hasn't been able to keep up with the demand of the ever-growing population [13].

On the other hand, global food demand has been increasing from COVID-19 and the Russia-Ukraine war has further increased that [31]. Almost 90% of wheat and other grains are shipped from Ukraine, the "breadbasket of the world" but due to the war, the crops could not leave Ukraine. Thus, the global wheat demand is increasing. In addition, the crop fields that were destroyed during the war will not get us food in the next harvesting season, which will increase the global food shortage even more. As India is one of the countries Bangladesh imports wheat from, the restriction on wheat export from India escalates the wheat demand problem for Bangladesh.

Lastly, there are several wheat diseases and most of them have been dealt with in various cultivars. However, Bangladesh witnessed the Wheat Blast disease for the first time in February 2016 which was caused by MoT [15]. Wheat production decreased to 0.349 million acres in the 2017–2018 growing season, which was the lowest level in the previous three decades. As a result, the Bangladeshi government advised farmers to abandon wheat in favor of other crops including lentil, boro rice, maize, etc. and this led to a 52% decrease in the areas used for wheat growing. All these reasons are causing a reduction in wheat production and farmers are afraid to cultivate wheat. This is why it is really critical to detect diseases at an early stage or even correctly so that the little amount that's being cultivated can be harvested in full.

1.1 Problem Statement

There have been many research works until now, where there are some works with SVM, VGG16, ResNet50, InceptionV3, EfficientNet and many other CNN models. In addition, few of the authors tried to get a higher percentage in the accuracy rate using some hybrid models. Some of the models did fascinating works for a few wheat disease detection with great accuracy. However, most of the research conducted till now had a limited number of images for training the models and those models could be more efficient and robust by using a larger dataset. Again, some deep CNN models are heavy on resources, which could be improved in the future by fine-tuning parameters or by reducing the number of fully connected layers. There are papers with wheat diseases, but there have very little research on wheat blast.

In this study, we intend to work on a private dataset that incorporates one unique wheat disease. We paid particular attention to the Wheat Blast, which, if not discovered quickly, might harm the entire field. In addition, we included more photographs of Wheat Blast, which, as far as we are aware, has never been done before. Our proposed ensemble model will be trained on our custom dataset.

1.2 Research Objectives

With the use of CNN, we hope to create a deep-learning model and make an ensemble model with ViT, that can identify various wheat diseases such as Wheat Blast, Leaf Rust, Yellow Rust, Septoria etc. Our proposed ensemble model will check for the presence of an infected plant or leaf after completing all necessary preprocessing. There have been several CNN models as well as some hybrid ones that have performed well, but as far as our knowledge goes, there's few with the disease Wheat Blast. After conducting this study, we look forward to:

- Have a solid understanding of the Deep Learning model and AI.
- Develop a model that will detect whether a wheat leaf is diseased or not so that wheat cultivation can be benefitted.
- Compare the proposed ensemble model with other deep learning models.

1.3 Thesis Structure

To begin with, chapter 1 starts with the Introduction. Secondly, the Problem Statement for this paper is stated in subchapter 1.1 while the subchapter 1.2 includes

Research Objectives and Thesis Structure is provided in subchapter 1.3. Thirdly, Background is covered in chapter 2. Following this, Existing Deep Learning Models are described in subchapter 2.1 and Literature review is included in subchapter 2.2. After that, chapter 3 discusses the Dataset where Data collection, Data Analysis, and Data Preprocessing are covered in subchapters 3.1, 3.2, and 3.3 respectively. In addition, subchapter 3.3.1 describes a few examples of the dataset's photographs. Consequently, the presentation of Methodology follows in chapter 4 whereas, subchapters 4.1 and 4.2 cover Ensemble Models and Proposed models, respectively. After that, chapter 5 includes Result and Discussion. Finally, a conclusion is presented in chapter 6.

The following workflow is maintained in this study and explained in different chapters.

Workflow

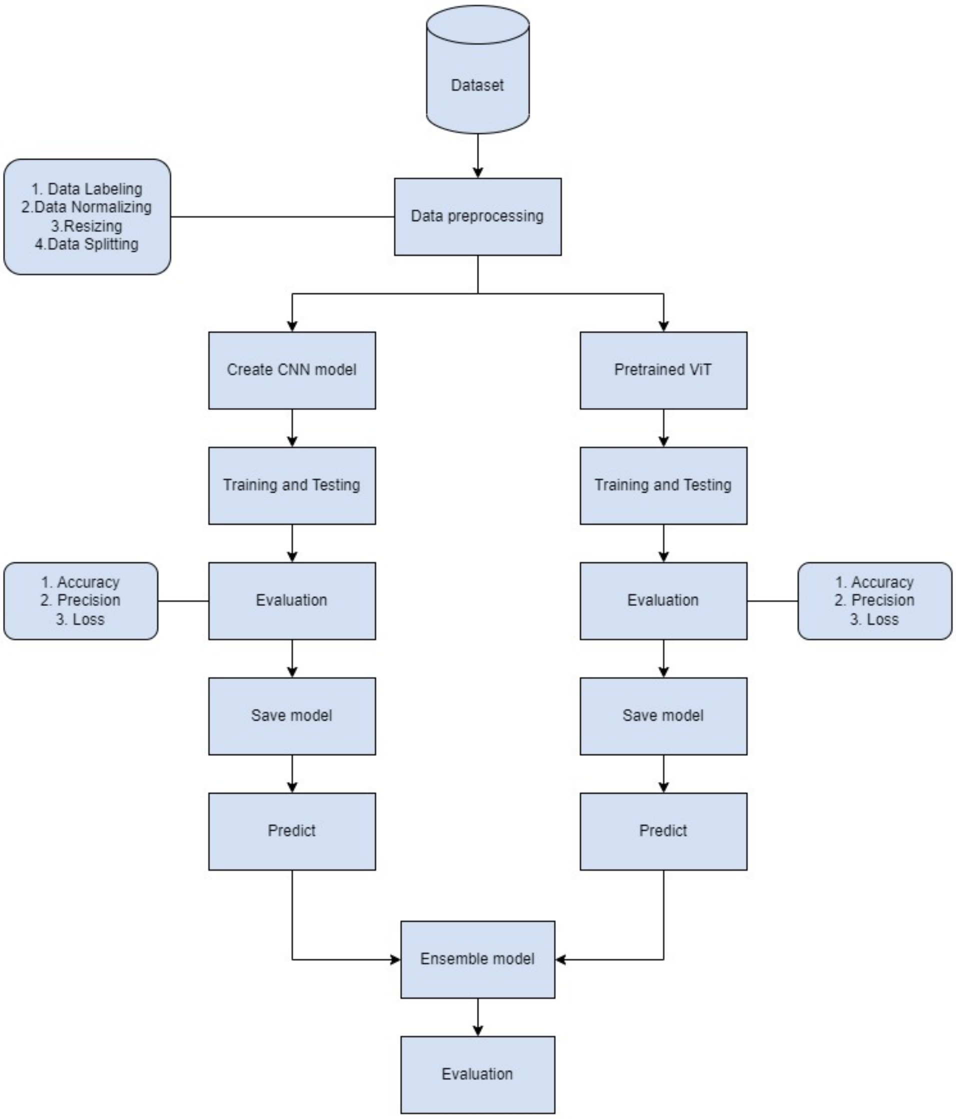


Figure 1.1: Approach of the proposed model

Chapter 2

Background

2.1 Existing Deep Learning Models

Deep learning algorithms have been used extensively in research up to this point. Among these, NN has been used along with support vector machines (SVM), SVM, and NN achieving an accuracy of 89.23% and 80.21% respectively [8]. An enhanced VGG16 model that outperformed the single-task learning approach was suggested [25]. Then, a CNN was employed to categorize the three wheat illnesses of Powdery mildew, Yellow rust, and Stem rust [11]. A pre-trained CNN using GoogLeNet architecture was used, where the overall accuracy was 94% [14]. Another proposed model got a testing accuracy of 97.88%, representing improvements of 7.01% and 15.92% over VGG16 and RESNET50, respectively [24]. Furthermore, between AlexNet and GoogLeNet, GoogLeNet performed better, having an accuracy of 99.35% [7]. Another model that was evaluated using images from the Sukkur IBA University performed with an accuracy of 98% and was trained using 38 classes from the PlantVillage dataset [18]. A different image processing method used crucial programs such as GLCM, PNN, and k-mean clustering [9]. The VGG-FCNVD16 model trained on the WDD2017 database showed an accuracy of 98.84% [5]. Improved Inception-v3 and ResNet-152 have a corn crop accuracy of 97.81% and 97.48%, respectively [26]. Another study used DL and image processing techniques to precisely identify ill regions in color photos of wheat spikes [16]. Many labeled pictures were split into training and validation datasets in order to retrain a Mask RCNN model learning technique to identify and detect spikes in images. The sick areas of the spikes in the grayscale images were suggested to be highlighted using a novel color feature called GB. The enhanced region-growing algorithm successfully identified the FHB-infected regions of each spike. The results showed that the RG algorithm outperformed K-means and Otsu's method for segmenting ill regions. Additionally, utilizing Random Forest, Partial Least Squares Regression, Support Vector Regression, and CNN to compare the spectra and images from FHB-infected wheat heads, a more generic model based on hyperspectral imaging was developed [20]. A different hybrid model, which makes use of a larger testing dataset, has the best generalization performance for F1 score and accuracy, which are 0.75 and 0.743, respectively [12].

There are pre-trained models such as VGG16, AlexNet and ResNet50 which have been trained on a different dataset but performing really well in the case of a different dataset. Thus, our proposed CNN model will be detecting all the wheat diseases there are as no one else has done this before as far as our research goes. This will be beneficial to our farmers for detecting diseases with the same symptoms, multiple diseases at the same time or contagious diseases like a Wheat Blast from spreading and reduce the loss as it will be faster than traditional laboratory test methods and less labor-intensive [6].

2.1.1 CNN

For image classification and image recognition, CNN is used. For scene labeling, face recognition, and objects detections, CNN is the most well known model. It gives good performance because of convolution operation. Convolution operation is responsible for identifying the feature and edges from images. In Convolution operation, an image's resolution determines how the computer interprets it as an array of pixels. With reference to the image resolution, it will see as height *width *dimension.

To process each input image in a CNN network, a number of convolutional layers along with fully connected layers, pooling layers and filters are used. After that, an object will classify between 0 and 1 using the Soft-max function.

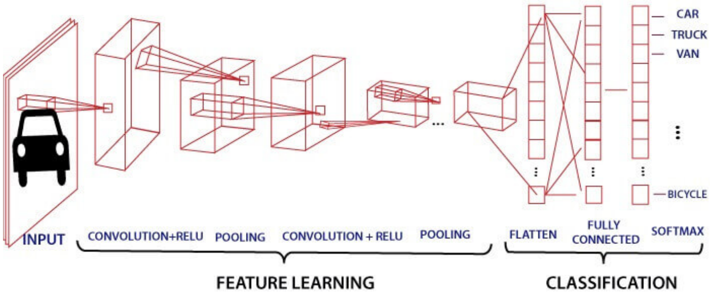


Figure 2.1: The architecture of CNN

The stride is the amount of pixels that have been displaced across the input matrix. When the stride is equal to 1 the filters are being moved one pixel at a time , and similarly, two pixels each time when the stride is uniform to 2. Padding is the concept of adding layers of 0 around an image.

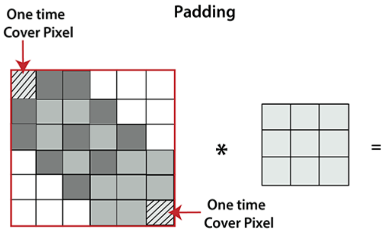


Figure 2.2: Padding

The corner pixel will only be covered once, as shown in the Figure 2.2, but the central pixel will be covered multiple times. It indicates that we know additional details

about the middle pixel. To overcome this, we introduce padding. The pooling layer is responsible for the reduction of the number of parameters when photographs are too large. ”Downscaling” the image created from the previous layers - this is the pooling process, which is similar to scaling down an image that helps decrease its pixel density.
Max Pooling’s main objective is to reduce the dimensionality of an input representation, in order to make assumptions about the features present in the sub-region binned. Using a max filter on non-overlapping subregions of the initial representation is the process to accomplish max pooling.

Average pooling is a technique for downscaling that divides the input into rectangular ”pooling zones” and calculates the average values for each one. Mean Pooling is the sub-region that is configured precisely the same for sum or mean pooling as it is for max pooling, but sum or mean is used in place of the max function.
The input from the other levels is sent after being flattened into a vector in a totally interconnected layer, which is a completely linked layer. The output will be changed into the desired number of classes by the network.

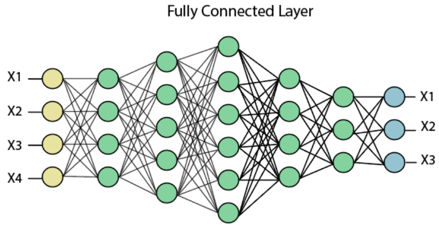


Figure 2.3: Fully Connected Layer

Using completely linked layers, the features matrix map in the image above is transformed into a vector, such as x1, x2, x3, and so forth. To identify the outputs as a car, dog, truck, etc., we will combine features to build a model and then apply an activation function like softmax or sigmoid.

There are many advantages of CNN. Firstly, it requires less preprocessing than alternative classification models because they can automatically learn hierarchical feature representations from the unprocessed input images. It also has a high accuracy rate. The disadvantages of CNN are its computational requirements are high. It can create difficulty with small datasets.

2.1.2 Inception V3

It has 48 layers. This model is a CNN, which can load a network that has already been trained using more than a million photos from the ImageNet database [33]. Label resizing is a feature of the Inception architecture. A few enhancements include batch normalization for layers in the sidehead, factorized 7 x 7 convolutions, and an extra classifier to move the label information lower down the network.

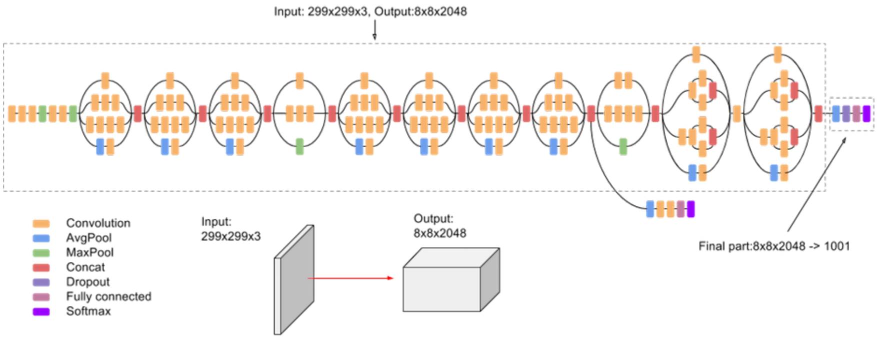


Figure 2.4: The architecture of Inception V3

The architecture of the InceptionV3 network is built in this way. To begin with, factorized Convolution reduces the amount of parameters employed in a network, which improves computational efficiency. It also monitors the network's effectiveness. Smaller Convolutions therefore aid in accelerating training by substituting smaller convolutions for bigger ones.

This well-known image recognition model, Inception v3, has been shown to attain more than 78.1 percent accuracy on the ImageNet dataset and nearly 93.9 percent accuracy in the top 5 results. The model is the outcome of multiple ideas that different researchers have developed over time.

The model is like a puzzle made of different pieces. These pieces can be symmetrical (balanced) or asymmetrical (not balanced). Batch normalization, which is also used for the activation inputs, is heavily utilized by the model. Using Softmax, the loss is calculated.

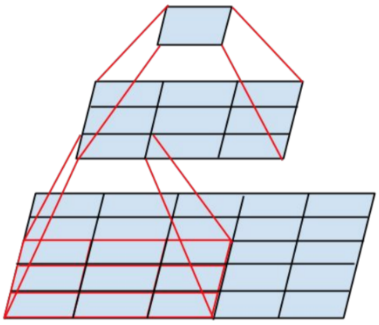


Figure 2.5: 3×3 convolution to fully connected layer of Inception V3

In the above Figure 2.5, fully-connected layer with 3x3. Since both 3x3 convolutions can share weights among themselves, the number of computations can be reduced.

A 1x3 convolution followed by a 3x1 convolution could be used in place of a 3x3 convolution in asymmetric convolutions. If a 3x3 convolution were replaced by a 2x2 convolution, the parameter numbers would be greater than the suggested asymmetric convolution. Furthermore, the extra CNN classifier added between layers during training and loss adds to the loss of the base network is a CNN that is used to the addition for the original network. GoogLeNet uses auxiliary classifiers for a deeper network, as opposed to Inception v3, which uses them as a regularizer. Finally, The size of the grid is often decreased via pooling activities.

2.1.3 ResNet

The ”vanishing gradient” problem, in which gradients get smaller as they move through the layers and make it difficult to train the network’s lower layers, is one of the main difficulties in training very deep neural networks. To address this, ResNet introduced a cutting-edge technique called ”residual learning,” which enables the construction of much deeper networks by more efficiently facilitating information flow through the layers.

In a deep learning model known as a residual neural network, the weight layers train residual functions depending on the inputs from the layers. A network with skip connections that perform identity mappings and are added to the layer outputs is referred to as a residual network. Deep learning models having tens or hundreds of layers were able to train quickly and get closer to higher accuracy as they added more layers thanks to residual networks.

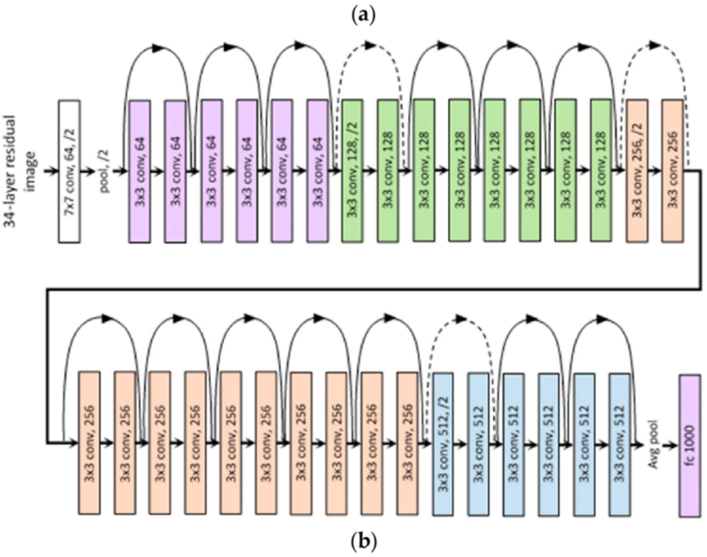


Figure 2.6: The architecture of ResNet

Typically, ResNet architectures are composed of a number of ”residual blocks”, each of which has a number of convolutional layers with oblique connections [30]. Each block’s initial convolutional layer shrinks the input’s spatial dimensions, while each block’s final convolutional layer enlarges them to their original size. ResNet can build far deeper networks while still preserving accuracy by stacking these pieces together.

Among the 3 basic parts, a set of convolutional layers that perform feature extraction make up the initial part. Batch normalization and an activation function, usually ReLU, are then applied to add non-linearity. The second element is a skip connection, which only sends the input to the block’s output. It is referred to as the

identity block. Another set of convolutional layers make up the third component, which is then followed by batch normalization and an activation function. The input that was received through the skip connection is combined with the output of this component.
It's important to remember that the precision of ResNet models can differ depending on elements like the caliber and diversity of the training data, the preprocessing methods used, the training optimization algorithms used and the hyperparameters selected. Deeper ResNet models typically have higher accuracy but demand more computing power during training.

2.1.4 EfficientNet

EfficientNet is a CNN that enhances the performance of an existing ConvNet based on the available resources by using a scaling technique called compound scaling. (memory and FLOPS). Scaling a ConvNet refers to the adjustment of the network's dimensions for acquiring better performance, the dimensions of a ConvNet are; depth, width and resolution.

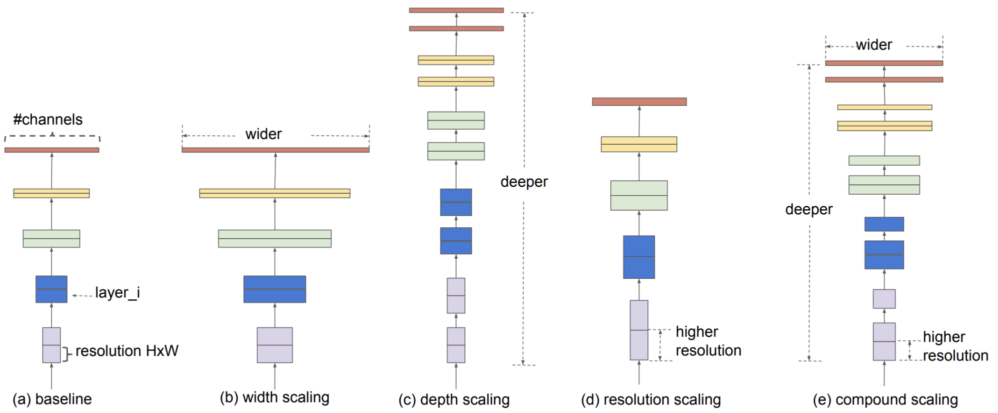


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Figure 2.7: Different types of Model scaling

Depth is the quantity of network layers, width is the number of channels in the convolutional layer, and resolution means the input images width×height. EfficientNet mainly proposes compound scaling. Compound scaling refers to the process of uniformly scaling each dimension of the network (width, depth, and resolution) by a fixed ratio in order to achieve superior scalability. So, network's dimensions are Depth: d $= \alpha^{\phi}$, Resolution: r $= \gamma^{\phi}$, Width: w $= \beta^{\phi}$. Here, a small grid search can be used to obtain the constants α, β and γ, and ϕ is referred to as the compound coefficient.

The user can regulate the quantity of resources available by specifying a coefficient called ϕ. The network's depth, width, and resolution are given to these resources, respectively, by α, β and γ. The Compound scaling extends the network's dimensions. Total number of floating point operations required for a single forward pass is FLOPs.

In order to achieve optimal accuracy and FLOPS, EfficientNet-B0 employs a multi-objective neural architecture. The aforementioned design is based on the mobile inverted bottleneck MBConv, also known as the inverted residual block with an additional SE (Squeeze and Excitation) block.

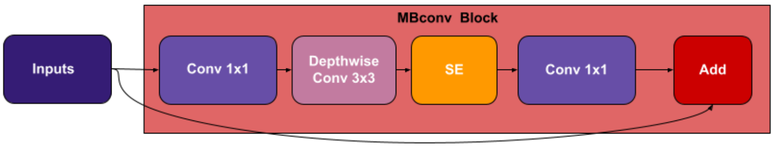


Figure 2.8: The Architecture of MBConv block

The compound scaling method was used to scale up from the baseline network EfficientNet-B0 [17]. The majority of a Convnet's computing expenditures are incurred during convolution processes, hence applying compound scaling elevates its FLOPS by $(\alpha.\beta^2.\gamma^2)^\phi$, therefore, $\alpha.\beta^2.\gamma^2 \approx 2$, to increase the total FLOPS by 2^ϕ. So, at first the ϕ is set to be 1 and a grid search method is applied to find the parameters α, β and γ based on the equation and under the constraint. The results were α=1.2 , β=1.1 and γ=1.15.
After that, the new parameters are to be fixed and to acquire a family of neural networks EfficientNet-B1 to B7, the network's dimension equation were applied. The obtained family of neural networks is known as EfficientNet.

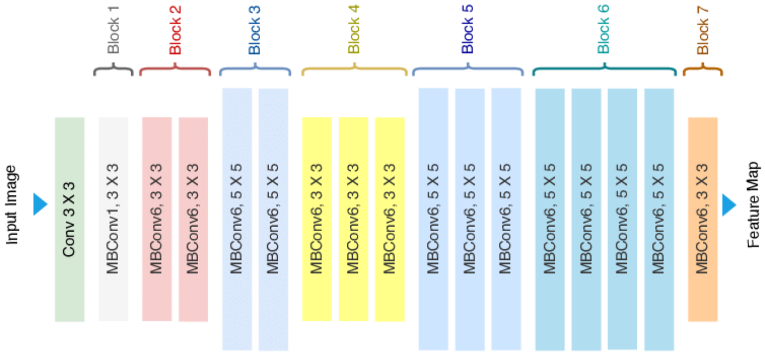


Figure 2.9: The Architecture of EfficientNet

As image size increases training becomes slow, in order to accommodate large images on the GPU, the batch size needs to be decreased here. But batch norms don't work well with small batches, and also training becomes sub-optimal with small batches. Also, it has depth wise convolutions, which is usually slower.

On the CIFAR-100, Flowers and three other transfer learning datasets, the EfficientNets perform exceptionally well and attain state-of-the-art accuracy of 91.7% (CIFAR-100) and 98.8% (Flowers) respectively with significantly fewer parameters.

2.2 Literature Review

The authors have collected the images first, then they preprocessed the images, after that they did the image segmentation process, and lastly they did feature extraction which included color, shape and size [8]. For classification, they used neural networks with an accuracy of 80.21% and support vector machine achieving an accuracy of 89.23%. There isn't a unified image segmentation method for some of the prevalent wheat diseases because the diseases that affect wheat leaves are diverse [2]. The authors tried to solve it by using the K-means clustering algorithm for an automatic and efficient segmentation of color images because coloured photographs contain more information than grayscale ones. Wheat leaf diseased areas are automatically distinguished from healthy ones using the K-means clustering technique and then the image is translated from RGB to Lab color space, and the pixels in the latter are clustered according to how close the pixels are to one another. For some diseases of wheat— powdery mildew, stripe rust, leaf rust—this technique improves accuracy to above 90%.

The authors have used an image hashing algorithm to create a dataset of 2414 images where 80% were labeled as single-diseased, more than 12% were labeled as healthy plants and around 6% were labeled as multiple-diseased [22]. The method utilized for disease recognition is based on the EfficientNet-B0 neural network architecture. Among the 3 techniques they applied, the first one was transfer learning, the second one was grouping data for training, validation and testing samples, the last one involved the transfer and augmentation of image styles. Among these, transfer learning and augmentation performed with accuracy of 0.942 which was better than the second one.

They have classified the severity of wheat blast into 3 categories using DCNN [23]. They trained their model on 2 different datasets, one containing both premature and maturing spikes and the other containing only the premature spikes. When identifying the photos, the models trained on only premature wheat spikes displayed greater recall, precision, and F1 score than the models trained on both types. When a wheat spike matures the color changes to yellow/white which can confuse the CNN model. Again, FHB may confuse the model as well as their symptoms are quite similar (e.g. spike bleaching). This study has been conducted only on wheat blast and so the CNN model could be made more generalized by using a larger dataset and variety of diseases. The improved VGG16 model, trained on a small dataset, performs better than some methods including single-task learning, ResNet50, the reuse-model of transfer learning and DenseNet121 [25]. This multi-task transfer learning for recognizing the diseases of wheat leaf and rice is better in generalizing while it is able to reduce the number of parameters.

CNN has been used for classifying three wheat diseases which are Stem rust, Yellow rust, Powdery mildew [11]. Their dataset contained four classes, three of which were the diseases mentioned earlier, and the other one was healthy leaves. Each of the classes contained 2707 images. AlexNet was used for training the dataset. We are not getting the full potential of Deep learning techniques due to small numbers and less diversity of images [14]. This is why they proposed Data augmentation to increase the number of images and to increase the diversity they applied image segmentation to work with lesions and spots instead of the whole image, which helped them to get multiple conditions from a single image. They used a pre-trained CNN using GoogLeNet architecture where the overall accuracy they got is 94% by using individual lesions and spots, higher than the ones using original images. However, the model could be more accurate if there were more data.

The accuracy of a DL model varies depending on size, the class imbalance, and the variety of the dataset, so the authors have worked with not only leaves but also spikes and roots of a wheat plant in this paper [24]. They tested their suggested model with the pre-trained ResNeT50 and VGG16 deep learning architectures. The model has been trained using more than 12,000 photos of wheat illnesses, while all other models have been trained up to 1000 epochs. The model has classified wheat diseases of 10 classes of the LWDCD2020 dataset with a training accuracy of 98.62% and a testing accuracy of 97.88% which is 7.01% and 15.92% improvement against VGG16 and ResNet50 respectively. In order to address the low accuracy and poor stability for the prediction of wheat FHB, the author here provided a model that combines a logistic regression mechanism-based and KNN and created a remote sensing approach to predicting FHB [29]. The disease prediction variables and factor weights were utilized in the KNN model to predict the prevalence of FHB. The results demonstrate that without taking wind speed predictors into account, the logistic-KNN model can improve the stability and predictability of machine learning models. There will be many computations for the KNN model. In the future, they may implement k-d tree to simplify model's computations and increase the model's applicability.

Their goal was to classify crop species, presence of any disease and identity of the disease if there was any [7]. The dataset for classification was collected from the PlantVillage project and it contained 54,306 images among which the number of crop species and crop diseases were 14 and 26 respectively. Among the two compared CNN architectures which are AlexNet and GoogLeNet, the latter performed better both in terms of transfer learning and training from scratch. 993 images out of 1000 were correct when their model tried to classify crops and diseases, resulting in an accuracy of 99.35%. However, if the model was to identify any images that were taken under different conditions than the ones that have been used for training, the accuracy drops to 31%. The accuracy of the model can be improved by creating a more diverse dataset for training, where leaves may face downwards or diseases may occur at different parts of the plant. As detecting plant diseases by looking at leaves, stems or fruits is a job requiring human resources, the authors have used CNN to detect plant diseases from the PlantVillage dataset where they worked with 15% of the total images [18]. The model trained using 38 classes of the dataset was tested using the images of Sukkur IBA University, and it performed with an accuracy of 98%. There could be more actual environmental images, plant types and diseases for improving the accuracy of the model. In future the model may adopt a 3 layer approach where the first layer will be detecting plants in an image, the second layer will be telling the plant type and the last layer will be used for classifying diseases.

The paper is written based on India which is an agricultural country as well as Bangladesh [9]. Farmers are not aware of what types of diseases plants have and how to prevent these. Utilizing an image processing method to find plant diseases is one solution. Image acquisition, pre-processing of images, extraction of features, and finally, a classifier known as a neural network are included in this procedure. In order to identify leaf disease, a typical image is first captured using any digital camera, and the image quality is then improved through pre-processing. Afterward, the image is split into clusters, with the infected area remaining and the green pixels being masked. Segmentation is the technique used after the masking to increase precision. Feature extraction and statistical analysis are completed following segmentation. A neural network is finally used to classify the images. The image processing method used for this plant disease identification makes use of crucial programs including GLCM, PNN, and k-mean clustering. One of the difficulties is that the plant needs to be regularly watched in order to discover the illness in its early stages. In future work, a large data must be used for training purposes in a neural network. The increment of training data results in the accuracy of the system being high. And fuzzy logic can be used as a classification tool and the accuracy rate and speed of the system can be compared.

This work demonstrates the detection and classification of diseases that damage wheat harvests using a two-dimensional CNN model [28]. The report mentioned the use of VGG-FCNVD16 as a method to circumvent the issues and achieve an average production of 97.95 percent. They used WDD-2017 as the dataset. Similar image pre-processing on the acquired dataset, importing the required elements, opening the dataset and loading the photos, the ability to export binary images as pixel arrays after conversion, etc. After gathering 4800 photos in total for this study, the dependability of the results was determined to be 98.84 percent. designing software that can manage more types of diseases by adding new disease classes to its capability list and using multi-dimensional CNN could increase the accuracy of disease detection.

In this study, they extracted the pollutants from wheat using near-infrared (NIR) hyperspectral imaging [5]. Then, raw NIR reflectance spectra (in the NIR region of 1000-1600 nm) of these contaminants and wheat were gathered. With the use of SVM, Naive Bayes (NB), and k-nearest neighbors (k-NN) classifiers, the raw and pre-processed data were divided into categories. Two-way and multi-way classifications were used in each study to comprehend the categorization of each type of wheat contamination, as well as the classification of all wheat contaminant types.When pre-processing spectral data with the SNV technique and classifying the results with a k-NN classifier, contaminants and wheat had the highest classification accuracy. After applying different approaches the authors got the results that showed that canola was most accurately classified as a foreign material, while oats were the least accurately classified. The three classifiers (SVM, NB, and k-NN) all had accuracies of around 79.0±0.8%. The study suggested using ResNet-152 and Inception-v3 versions to identify diseases in essential crops like rice and corn and they both added layers to those models to improve accuracy [26]. Incorporating a new dense layer into both models improved overall disease classification and identification accuracy using the dataset of rice and maize leaves. For the corn crop, the suggested method's accuracy was 97.81% and 97.48%, respectively, using variants of InceptionV3 and ResNet152. More deep learning models, including NesNetMobile and NesNetLarge, as well as AlexNet, VGG and all of its variants, Xception, MobileNet, and DenseNet, are required to examine accuracy in this work. An expanded version of this research work can be used to investigate additional plant leaf diseases using the same methods.

This paper demonstrates how to detect FHB disease in wheat crops using DL and image processing methods [16]. Color images of wheat spikes taken at the milk stage were processed to create datasets, which are used for developing a reliable and cost-effective high-throughput phenotyping system (helpful for developing resistant wheat varieties). A Mask R-CNN model was retrained through the transfer learning technique to successfully detect and identify spikes in images. Only the last few layers of the network were trained, resulting in an accurate detection rate for most spikes across several wheat varieties with different morphologies and sizes. R2, RMSE, and rRMSE all remained at 0.80%, 1.17%, and 21.37% respectively when assessing the model's accuracy using testing data sets compared to manual counts alone. The mAP achieved was 0.9201. The sick areas of the spikes in the grayscale images were suggested to be highlighted using a novel color feature called GB. The FHB-infected regions of each spike were successfully detected by the improved region-growing algorithm. The segmentation of sick regions using the RG algorithm outperformed using K-means and Otsu's technique, according to the results. It has been shown that DL algorithms enable precise FHB detection in wheat based on color picture analysis, increasing the effectiveness of FHB assessment in the field. The results of testing 450 images showed that it is possible to detect FHB in wheat using color images.

The spectral and image of wheat spikes infected with FHB have been collected for comparing Support Vector Regression, Random Forest (RF), Partial least square regression and CNN for developing a more generalized model based on hyperspectral imaging [20]. The deep CNN (DCNN) model compared better than others as it scored 0.97 and 3.78 in R^2 and RMSE respectively. However, DCNN needs high hardware requirements and GPU support and more studies could be conducted to examine the impacts of different locations, years and wheat cultivars on the performance of the model since these factors may affect hyperspectral imaging and make the model more robust. The authors tried to detect FHB in early stages using hyperspectral imagery [12]. Instead of traditional methods, they have used a 2D data structure as the input for their CNN and then used a bidirectional recurrent and a convolutional layer for constructing a hybrid NN which would generalize the CNN model. The mode performed better than other deep learning models as it scored 0.75 and 0.743 in F1 score and accuracy respectively.

Improving the detection of mycotoxin deoxynivalenol (DON), formed by FHB, in wheat kernels due to the health hazards connected with its consumption is a primary research goal [3]. This study contrasted the speed of human work to the quicker process, and HSI is used as the basis for more reliable detection techniques. Here, two methods are used for FHB detection: the first uses images of wheat ears captured in the field, while the second uses images of kernels normally taken in a lab under controlled circumstances. First off, the created technique proved resistant to issues including form, orientation, shadowing, and clustering of kernels, with a classification accuracy of 91%. In addition to FHB detection, it was shown that the system could calculate DON concentrations in wheat kernels, albeit to a limited extent. Additionally, Delwiche, Kim, and Dong presented a study on the evaluation of Fusarium damage in wheat kernels using NIR and VR images in 2011 and with these, they used linear discriminant analysis (LDA) and obtained an accuracy of 95%.

In order to analyze the spectroscopic diagnosis of FHB, this study used continuous wavelet analysis (CWA) to examine the reflectance spectra of wheat ears (350 to 2500 nm) [19]. Hyper-spectral remote sensing has assisted in the identification of FHB in wheat ears. Additionally, employing CNN in a natural field, healthy and FHB-infected wheat could be distinguished. It's been noted that CWA did a good job at identifying crop stressors, monitoring diseases and pests. During the stages of wheat filling in 2018 and 2019, two separate hyper-spectral datasets in the range of 350-2500nm were used. The objectives of this work were to: (1) evaluate CWA's performance; and (2) determine the optimal wavelet features and both to detect FHB. An FLDA model was constructed using the six wavelet qualities as a base. The wavelet characteristics performed well in diagnosing FHB in winter wheat ears, as evidenced by an accuracy around 88.7% and a 0.775 kappa coefficient. The identified future plan included testing the practicality of CWA combining different spectral acquisition angles to find FHB in winter wheat at field scales using UAV hyper-spectral technology.

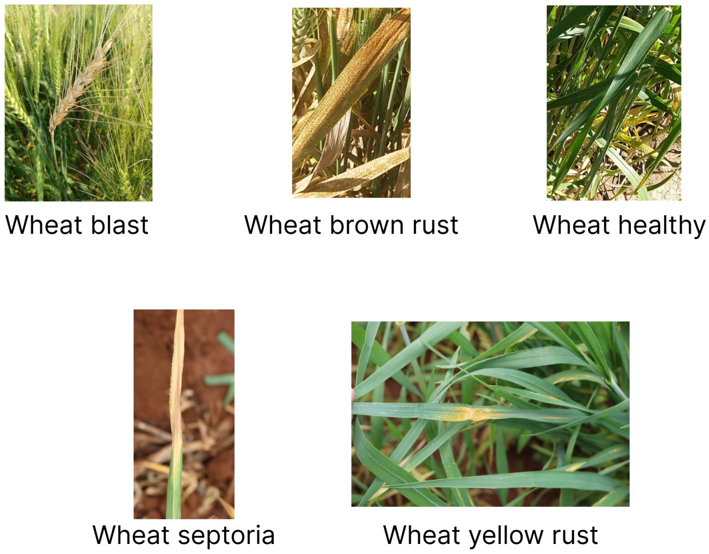
The author tried to identify wheat seeds from the TBIO Toruk cultivar that were FHB-infected using various deep-learning techniques based on RGB photos, integrating hyperparameter optimization and fine-tuning techniques with various pre-trained CNN [27]. The models detected FHB in seeds with a 99% accuracy and a 97% accuracy utilizing a low-complexity design architecture with hyperparameter optimization. These results point to the potential for accurate categorization of wheat seeds with FHB using low-cost imaging equipment and deep-learning models. Convnets were used to distinguish between sick seeds and healthy seeds using data collected from RGB digital photos in order to build deep-learning models for monitoring FHB in wheat seeds.The behavioral changes among wheat genotypes may affect the detection method's accuracy because FHB symptoms are genotype dependent.

Chapter 3

Dataset

3.1 Data Collection

The dataset contains wheat diseases such as brown rust, yellow rust, septoria and blast along with healthy wheat images. Wheat blast is collected from a private source. The images have been collected from Abu Noman Faruq Ahmmed (the Professor and Chairman of the Department of Plant Pathology, Faculty of Agriculture of "Sher-e-Bangla Agricultural University, Dhaka"), whereas the other diseases and healthy leaf images have been collected from multiple datasets on Kaggle and PlantVillage [4]. However, only the wheat related classes of PlantVillage dataset has been used here.



Wheat blast

Wheat brown rust

Wheat healthy

Wheat septoria

Wheat yellow rust

Figure 3.1: Images of different classes

3.2 Data Analysis

There were formerly 3800 photos among 5 classes. The table 3.1 shows that the wheat healthy leaf class contains the highest number of images which is around 1225 or 32.23% of total images. Sequentially, wheat yellow rust, wheat brown rust, and wheat blast contain 1132, 915, and 431 images respectively. Furthermore, the class wheat septoria contains the least number of images, which is 97 and that is only 2.55% of the total images. As a result, it can be seen there was an imbalance among the images of the classes and so data augmentation would be a good solution here.

Table 3.1: Previous Dataset analysis

Class name	**No. of Images**	**Percentage(%)**
Wheat Blast	431	11.34
Wheat Brown Rust	915	24.07
Wheat Healthy	1225	32.23
Wheat Septoria	97	2.55
Wheat Yellow Rust	1132	29.78
Total	3800	100

The table 3.2 shows the analysis of all 5 classes of our current dataset. The dataset contains in total 4535 images among 5 classes now. The number of images in the private class Wheat blast has decreased to 422, after filtering all the images that were collected. Now, it presents 9.31% of the whole dataset. In the brown rust class, it contains 1128 pictures, as more images have been found later, and its percentage on the dataset increased to 24.87%. Due to the addition of more healthy wheat spike or grain images, the number and percentage of the healthy class have also increased. Yellow Rust remains the same as before. Previously, there was an imbalance in the data because the Septoria class had too few images. To make a decent balance among the classes, data augmentation has been performed on this class only, resulting in an increment of images to 485 and the percentage to 10.69.%

Table 3.2: Current Dataset analysis

Class name	**No. of Images**	**Percentage(%)**
Wheat Blast	422	9.31
Wheat Brown Rust	1128	24.87
Wheat Healthy	1368	30.17
Wheat Septoria	485	10.69
Wheat Yellow Rust	1132	24.96
Total	4535	100

3.3 Data Preprocessing

The images of the classes were labeled at first, as images are the features and the labels are the output. As the NN models would be taking input images of a certain size, these images need to be resized before feeding to the NN models. Again, all the images have been normalized as the RGB values are in between 0 and 255.

Lastly, the images have been divided into train and test dataset with a ratio of 90:10. In addition, the train part of the dataset have been further divided into train and validation data which are used during the training of models. The ratio of train and validation data here is 70:30.

The images in the dataset were of different size or measurements initially. Therefore, to train the implemented models, the images have been resized into 224x224. Due to the constant image size, the computing load will be minimized, further improving the outcome.

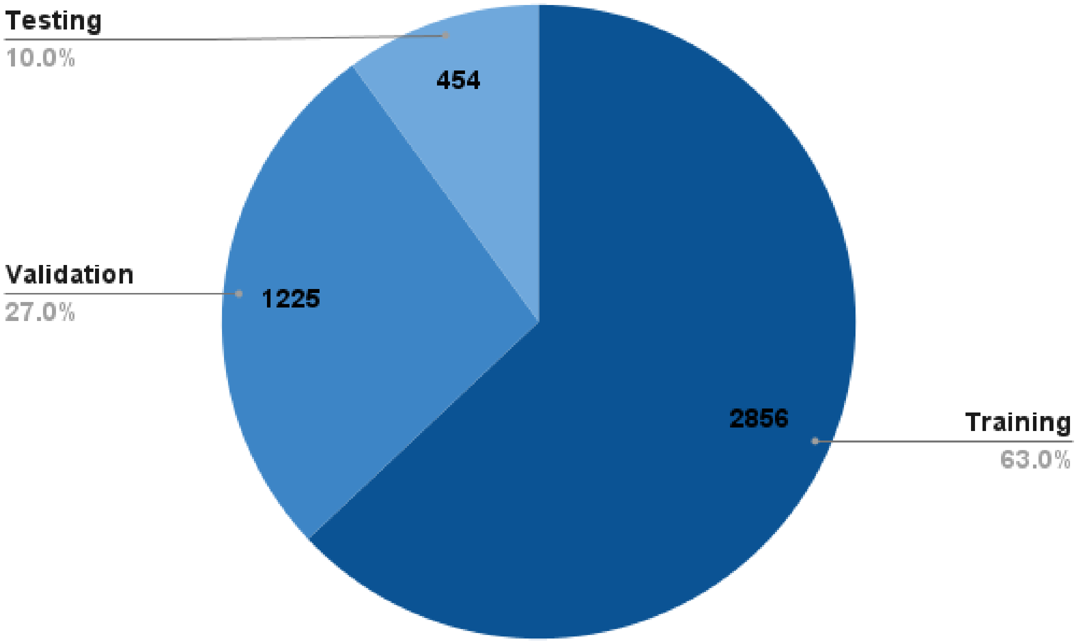


Figure 3.2: Train, Validation & Test image distribution

3.3.1 Sample Images from the Dataset

The below fig: 3.3 and fig: 3.4 shows the randomly generated Training, Validation images and their patches. In addition, fig: 3.5 shows the testing images.

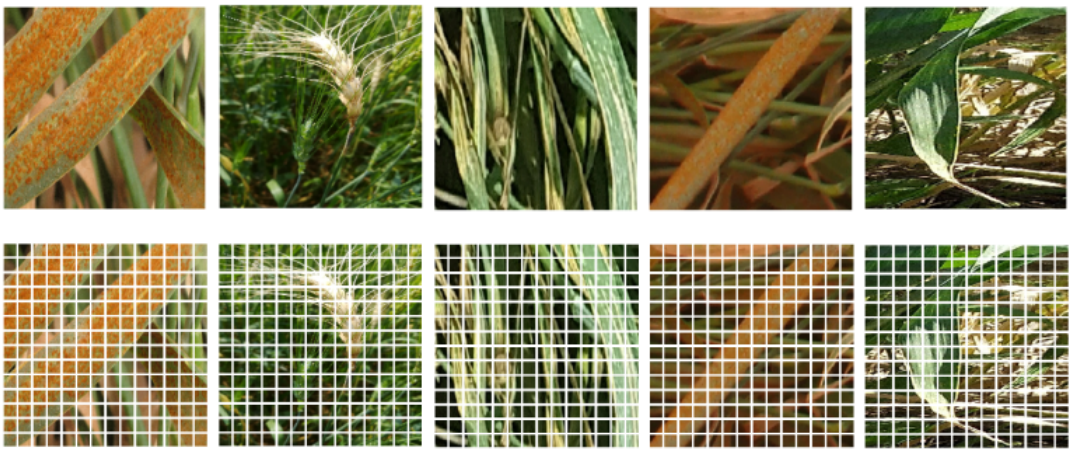


Figure 3.3: Random generated Training images and their patches

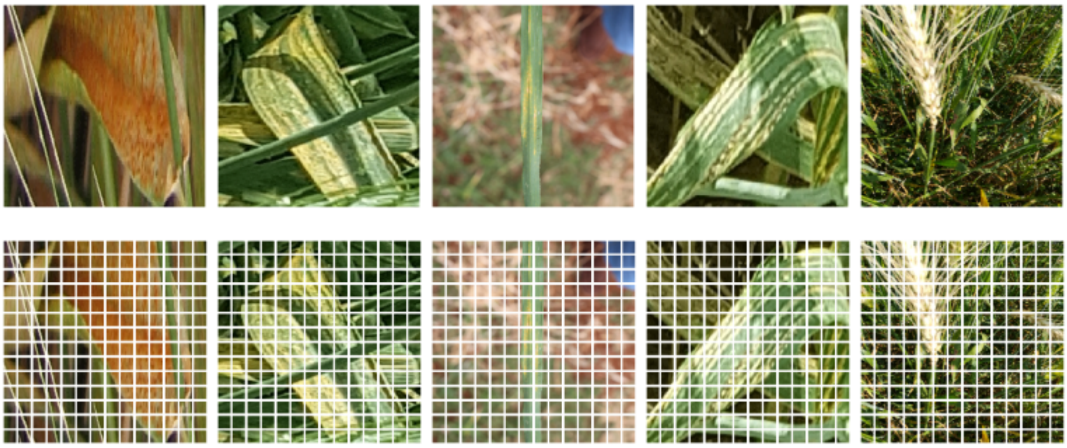


Figure 3.4: Random generated Validation images and their patches

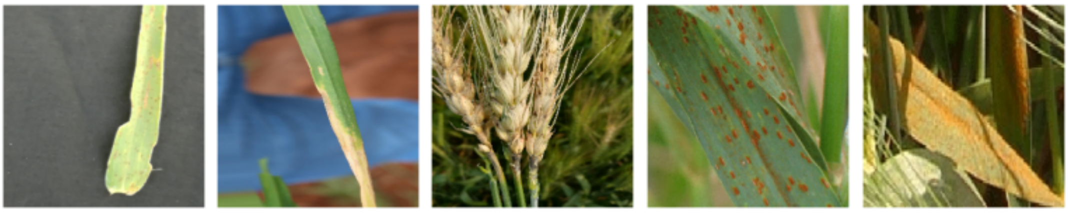


Figure 3.5: Random generated Testing images

Chapter 4

Methodology

4.1 Ensemble Models

The idea of ensemble learning is to integrate the predictions given by various machine learning models to create a more generalized model that has a higher prediction accuracy. There are mainly three types of ensemble learning methods, which are bagging, boosting and stacking.

Bagging involves predicting multiple samples of the dataset using multiple models and then averaging their predictions. Stacking is done by predicting different models on the same data and then using another model to get the best predictions. Boosting is implemented by passing a model's predictions to another model, and that one's to the next one. This is how a weighted average is achieved.

The deep learning models take a long time to train, and the results of the models are not guaranteed to be accurate even after training. The non-linearity of the neural nets can make the models sensitive to high variance. Thus, a solution to this problem could be training multiple neural network models, making them predict on the test data and then combining their predictions. This approach not only reduces the variance in the model but also makes the predictions more accurate.

All the pre-trained models used were mainly trained on ImageNet dataset for classifying 1000 classes. However, the number of classes is 5 for the wheat disease dataset and thus the top layers or the classification layers from the models have been removed, and custom classification layers have been added instead, to classify the 5 classes.

The classification head used for the pre-trained CNN models include a flattening layer to flatten the output from the feature extracting layers of the CNN models, followed by two dense layers having 128 and 64 neurons respectively with ReLU as the activation function. Then, the last dense layer with 5 neurons and softmax activation function has been added. The batch size is 32 and the optimizer used here is Adam with a learning rate of 0.0001.

On the other hand, the classification head used for the ViT model, has a BatchNormalization layer followed by three dense layers having 128, 64 and 32 neurons and ReLU activation function respectively. In addition, activation function softmax and five neurons have been used in the last layer for classification. The optimizer and the batch size are the same as those of the pre-trained models.

The pre-trained ViT model along with the classification head has the following layers:

Table 4.1: Layers with Parameters

Layers	**Output Shape**	**Parameters**
vit-b16(functional)	(None, 768)	85798656
flatten_1 (Flatten)	(None, 768)	0
batch_normalization	(None, 768)	3072
dense_3 (Dense)	(None, 128)	98432
batch_normalization_1	(None, 128)	512
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 32)	2080
dense_6 (Dense)	(None, 5)	165
Total params : 85,911,173		
Trainable params : 110,725		
Non-trainable params : 85,800,448		

4.1.1 ResNet50 & ViT

ResNet50, the pretrained model with a custom classification layer, has been implemented. Later, the ResNet50 model has been used to predict the test data. On the other hand, ViT has been fine-tuned on the dataset and again, it has been used to predict the test data. Finally, both of the predictions from the two pretrained models have been averaged. The workflow for this model:

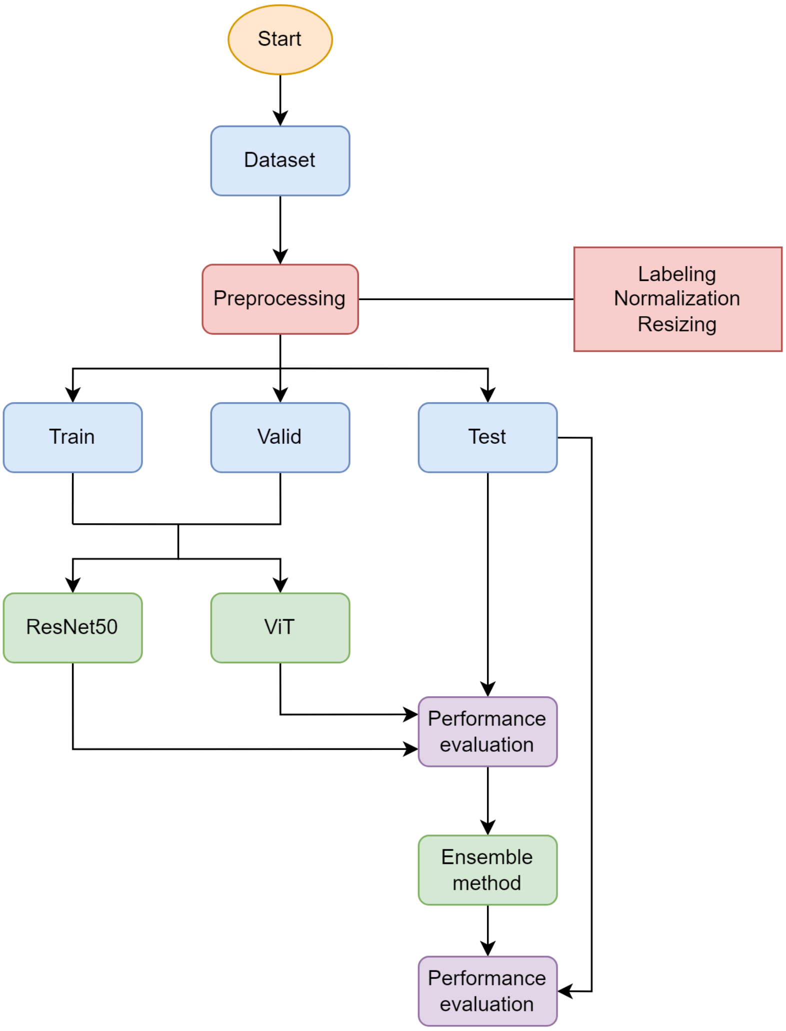


Figure 4.1: Ensemble model (ResNet50 & ViT)

4.1.2 EfficientNetB0 & ViT

EfficientNetB0, the first model of the EfficientNet family, has been implemented using a custom classification layer. Later, the pre-trained model has been used to predict the test data. Finally, predictions from the pre-trained EfficientNetB0 model, and the fine-tuned ViT have been averaged to create the ensemble model. The following workflow has been followed here:

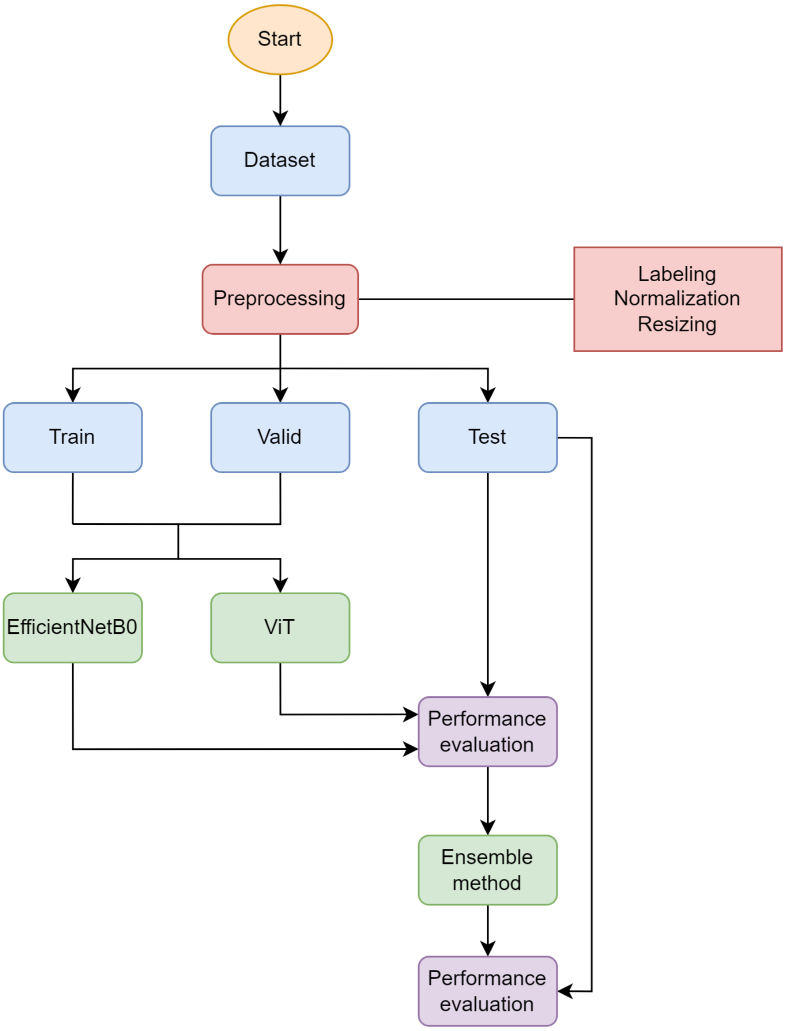


Figure 4.2: Ensemble model (EfficientNetB0 & ViT)

4.1.3 InceptionV3 & ViT

InceptionV3, the pretrained model with a custom classification layer, has been implemented. Consequently, the test data has been predicted by the model. Lastly, an ensemble model has been formed when both of the predictions from the pretrained InceptionV3 and the ViT have been averaged. The following diagram represents the workflow of the ensemble model:

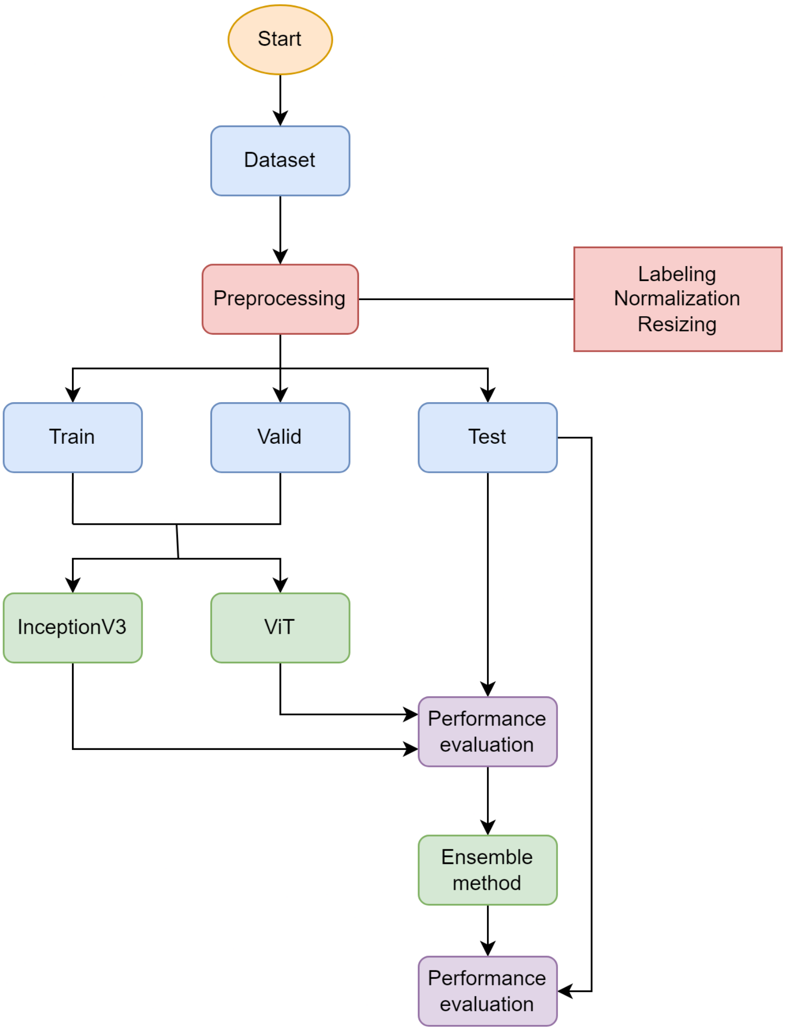


Figure 4.3: Ensemble model (InceptionV3 & ViT)

4.2 Proposed Model

4.2.1 Hybrid CNN

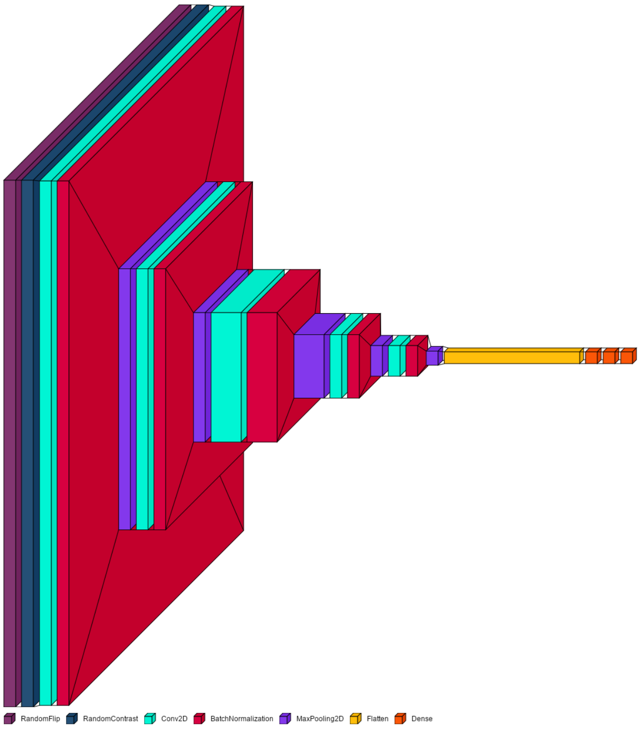


Figure 4.4: Convolutional Neural Network Architecture

The CNN architecture used in this research have two layers for randomly flipping images and creating contrast followed by 5 convolutional layers, where the first three layers have 64, 128 and 512 filters respectively and the last two layers have 128 and 64 filters. The filter size in the layers increased from 2×2 to 3×3 to 5×5. This increment in the size of filters helps CNNs to identify lines and edges in the first layers and keep increasing their focus area to identify bigger features with the increment in the number of layers. The same number of max pooling layers have been used, where the filter size and strides have been kept 2 throughout all the layers.

Then, after flattening the matrix, it has been passed onto a dense layer of 128 neurons which is connected to a dense layer of 64 neurons. Lastly, there's a layer with 5 neurons, as there are 5 classes in this dataset. The kernel initializer "He normal" has been used to initially set the weights of the filters for the convolutional layers.

In order to address the overfitting, batch normalization has been used after each convolutional layer and L2 regularization with a lambda value of 0.01 has been utilized in the hidden layers. Later, predictions on the test data have been done by the model.

From Figure 4.5 it can be seen that there was a good amount of overfitting at first and there are lots of curves. However, at the 50th epoch, the overfitting got reduced quite a lot.

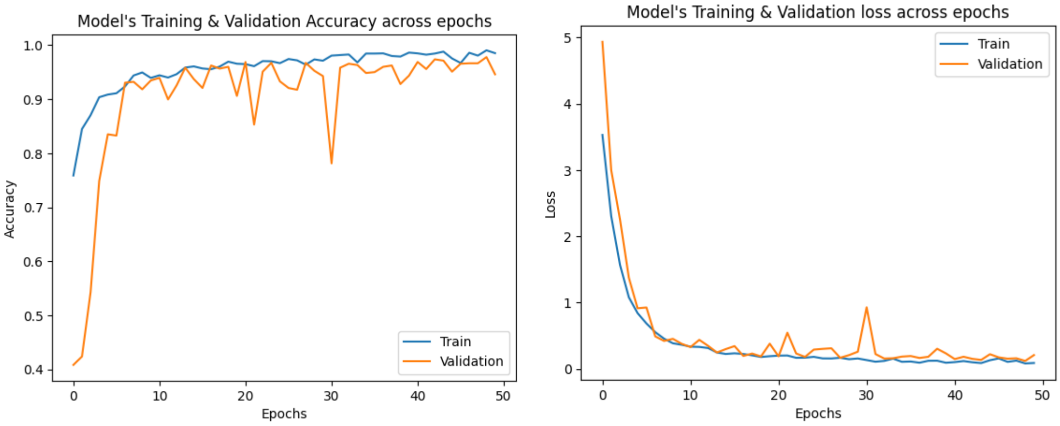


Figure 4.5: Accuracy and Loss curves of Hybrid CNN

On the other hand, the loss got reduced with training in training vs validation loss in Figure 4.5. After the 50th epoch both the values were close to each other even though the curves were a lot smoother compared to that of accuracy.

Table 4.2: Description of Layers and Parameters of the hybrid-CNN

Layers	**Output Shape**	**Parameters**
random_flip (RandomFlip)	(None, 224, 224, 3)	0
random_contrast (RandomContrast)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 223, 223, 64)	832
batch_normalization(Batch No)	(None, 223, 223, 64)	256
max_pooling2d (MaxPooling2D)	(None, 111, 111, 64)	0
conv2d_1 (Conv2D)	(None, 111, 111, 128)	32896
batch_normalization_1 (Batch No)	(None, 111, 111, 128)	512
max_pooling2d_1 (MaxPooling 2D)	(None, 55, 55, 128)	0
conv2d_2 (Conv2D)	(None, 55, 55, 512)	590336
batch_normalization_2 (Batch No)	(None, 55, 55, 512)	2048
max_pooling2d_2 (MaxPooling 2D)	(None, 27, 27, 512)	0
conv2d_3 (Conv2D)	(None, 27, 27, 128)	589952
batch_normalization_3 (Batch No)	(None, 27, 27, 128)	512
max_pooling2d_3 (MaxPooling 2D)	(None, 13, 13, 128)	0
conv2d_4 (Conv2D)	(None, 13, 13, 64)	204864
batch_normalization_4 (BatcH No)	(None, 13, 13, 64)	256
max_pooling2d_4 (MaxPooling 2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 128)	295040
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 5)	325
Total params: 1,726,085		
Trainable params: 1,724,293		
Non-trainable params: 1,792		

4.2.2 Vision Transformer (ViT)

ViT achieved highly competitive performance in object detection, image classification, and semantic image segmentation. It represents the cutting edge of image classification. It doesn't have any novelty over the Transformer, it's exactly the encoder network of the transformer.

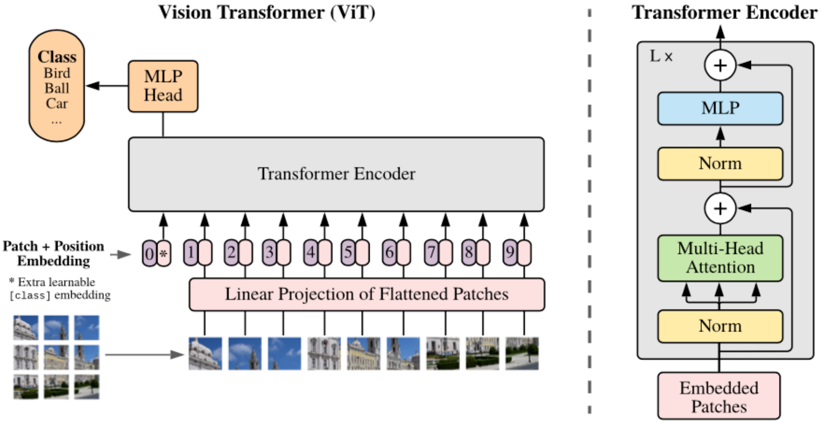


Figure 4.6: The architecture of Vision Transformer

A key component of computer vision is image classification, which entails giving an image a label depending on the content of the image. DCNNs, such as YOLOv7, have long been the state-of-the-art approach for classifying images.

Given that it was developed for NLP, the basic Transformer model only accepted a one-dimensional sequence of word embeddings as input. In contrast, the input data to the Transformer model is provided in the form of two-dimensional images when it is used to perform the task of image classification in computer vision.

There are smaller 2-dimensional patches created from the input image. C stands for the number of channels, whereas P stands for the patch size, in the input image of (H×W) resolution. This produces $K = \frac{H \times W}{P^2}$ number of patches each of which has a resolution of (P×P) pixels.

In all of its layers, the Transformer keeps a latent vector size of D. Therefore, a trainable linear projection is applied to the patches to flatten them into a vector size of D dimensions which produces a set of embedded image patches, also referred to as patch embeddings.

A learnable class embedding is prefixed to the list of embedded image patches. Its value indicates the categorization outcome, y. A final classification head is introduced during tuning after an initial classification head is attached to the state. Position embeddings were added to the patch embeddings in order to help the model to gain knowledge about the sequence ordering of the input tokens. The classification head is implemented in an MLP with a single hidden layer and a single linear layer, respectively, during pre-training and fine-tuning.

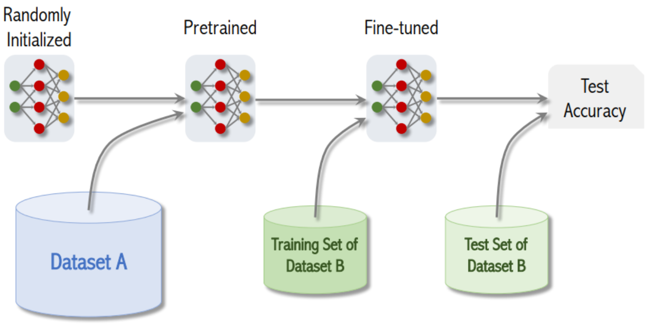


Figure 4.7: Working mechanism of ViT

Globally embedding information over the entire image is possible thanks to ViT's self-attention layer [10]. The model also learnt from training data, in order to encode the patches.

The Multi-Head Self Attention Layer's specification states concatenated to the dimensions linearly. To build the local and global dependencies of an image, several focus points helps here. MLP layer includes a pair of layers containing GELUs (Gaussian Error Linear Units). Since it doesn't introduce extra dependencies among training images, Layer Norm is inserted before each block. As a result, training time and performance as a whole are improved.

ViT can beat CNNs, especially when trained on huge datasets with over 14 million images [33]. It's like having a super expert for recognizing things in pictures. The best option is to use ResNet or EfficientNet if not. Simply removing the MLP layer and substituting a new layer with the formula D times KD*K, where K is the number of classes in the short dataset, will make all the necessary changes. The pre-trained position embeddings are represented in 2D in order to be changed in higher resolutions. This is done so that the positional embeddings can be modeled by the trainable liner layers [21].

The model also learnt from training data, in order to encode the patches. The Multi-Head Self Attention Layer's specification states concatenated to the dimensions linearly. To build the local and global dependencies of an image, several focus points help here. Since it doesn't introduce extra dependencies among training images, Layer Norm is inserted before each block. As a result, training time and performance as a whole are improved.

That is why in this research, transfer learning is used by adding the classifier head at first and then fine-tuning the model. After that, the model is tested on the test images of the custom dataset here.

Proposed Ensemble model

The predictions that have been calculated by the hybrid CNN and the predictions by the ViT model have been combined using the Averaging or Voting approach to form the basis of the suggested model for the study.

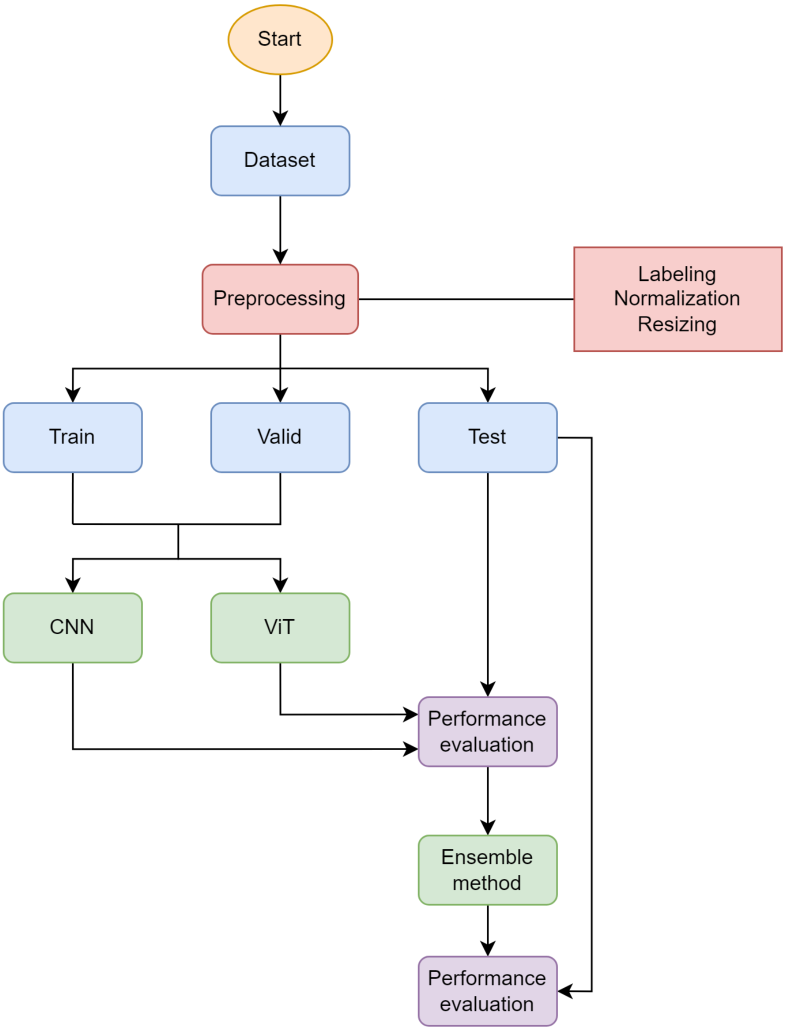


Figure 4.8: Proposed Ensemble model (Hybrid CNN & ViT)

Chapter 5

Results and Discussion

5.1 Experimental Setup

Table 5.1: Experimental setup for the training models

CPU	Intel 12th Gen Core i7-12700K
GPU	NVIDIA - GeForce RTX 3070 Ti 8GB
RAM	32 GB
Storage	1 TB

To ensure that any comparisons between the models are fair, they have all been developed and evaluated in the same experimental setting which are shown in the table 5.1. Additionally, these models are trained over 50 epochs. Here, the batch size and image size are specified and consistent across all models. Except for the proposed CNN, which has a slightly higher learning rate because it was developed from scratch, the pre-trained model's learning rate should be very low. In the table 5.2, the training hyper-parameters are shown.

Table 5.2: Hyperparameters used for the training models

Optimizer	Adam
Learning rate (Proposed CNN)	0.01
Learning rate (Pre-trained)	0.0001
Batch Size	32
Epochs	50
Image Size	224 × 224

5.2 Results

In this work, a custom dataset (privately and publicly obtained) has been used to train and test 5 distinct models. EfficientNetB0, ResNet50, InceptionV3, Vision Transformer (ViT) and hybrid CNN are the models that were put into practice. On the aforementioned dataset, these models were separately trained and tested. However, the various models are then individually ensembled with ViT. Among the ensemble models, the one which is based on the ViT and the hybrid CNN model, produced the best results when the results of all nine models were compared: EfficientNetB0, InceptionV3, ResNet50, ViT and the ensemble models.

Accuracy is like a score that shows how often a model is right in its predictions. It's the number of correct predictions divided by all the predictions it makes. So, if a model is accurate, it means it's doing a good job in making the right predictions.

Precision helps us figure out how many of the things we said were positive are actually correct. To find precision, we add up all the times we were right about something being positive (True Positives or TP) and divide it by the total of what we said was positive, whether we were right or wrong (True Positives + False Positives or TP + FP). This helps us know if our positive guesses are really accurate. The formula for Precision is:

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

The recall rate is like a measure to see how good we are at finding things. To figure it out, we take the number of times we correctly found something (True Positives or TP) and divide it by all the times that thing was actually there, even if we missed it sometimes (True Positives + False Negatives or TP + FN). This helps us know how accurate we are in detecting what we're looking for.

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

The F1 Score is like a grade that tells us how well a machine learning program is doing. It takes into account both accuracy (how often it's right) and recall (how good it is at finding things). It's like combining two scores into one to give us a better overall measure of performance. So, if a program has a high F1 Score, it means it's doing a good job in both accuracy and finding things.

$$F1\ score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5.3}$$

5.2.1 ResNet50

Table 5.3: Classification report of ResNet50

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.87	0.68	0.76	38
Wheat Brown Rust	0.57	0.78	0.66	119
Wheat Healthy	0.94	0.73	0.82	134
Wheat Septoria	0.53	0.24	0.33	33
Wheat Yellow Rust	0.72	0.78	0.75	130
Macro avg	0.73	0.64	0.67	454
Weighted avg	0.74	0.72	0.72	454
Accuracy			0.72	454

From the table 5.3 it can be seen that the pre-trained ResNet50 predicts the healthy class the most accurately as the F1 score for this class is 0.82. On the other hand, the model performs poorly in case of the septoria class, as the F1 score for this class is only 0.33. This might be due to the fact that there are only 33 images of septoria in the test dataset. The weighted average of the model is 0.72. The confusion matrix of the above-mentioned model is shown below:

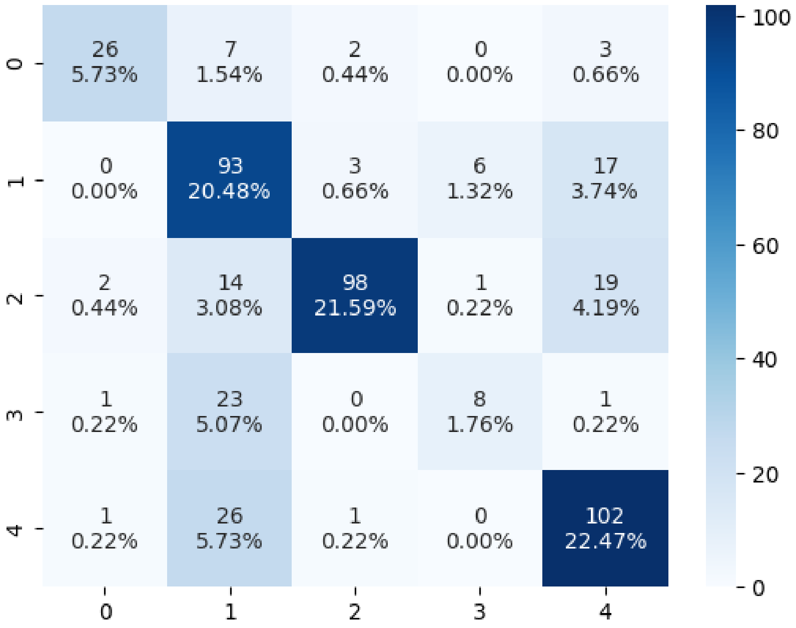


Figure 5.1: Confusion Matrix of ResNet

The confusion matrix 5.1 shows that the most number of images ResNet50 classified successfully are of the yellow rust class. However, the model has misclassified many images of this class and thus, healthy class got the most F1 score.

5.2.2 EfficientNetB0

Table 5.4: Classification report of EfficientNetB0

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.00	0.00	0.00	38
Wheat Brown Rust	0.30	1.00	0.47	119
Wheat Healthy	0.98	0.46	0.62	134
Wheat Septoria	0.00	0.00	0.00	33
Wheat Yellow Rust	0.00	0.00	0.00	130
Macro avg	0.26	0.29	0.22	454
Weighted avg	0.37	0.40	0.31	454
Accuracy			0.40	454

The table 5.4 shows the classification report of EfficientNetB0 on the custom dataset, where the yellow rust class along with blast and septoria, got a F1 score of 0.0 even though yellow rust had 130 images for testing. Due to this reason, the weighted average score is 0.31. The confusion matrix of the model is shown below:

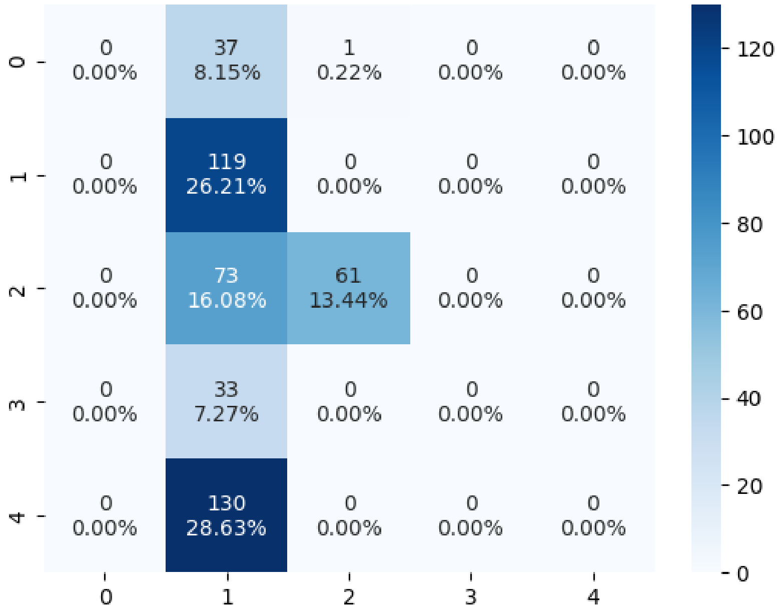


Figure 5.2: Confusion Matrix of EfficientNetB0

The classification matrix 5.2 shows the EfficientNetB0 model classifies the healthy class images the most successfully, making the F1 score of the class 0.62.

5.2.3 InceptionV3

Table 5.5: Classification report of Inception

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.89	0.89	0.89	38
Wheat Brown Rust	0.93	0.94	0.93	119
Wheat Healthy	0.95	0.95	0.95	134
Wheat Septoria	0.91	0.97	0.94	33
Wheat Yellow Rust	0.94	0.92	0.93	130
Macro avg	0.93	0.93	0.93	454
Weighted avg	0.93	0.93	0.93	454
Accuracy			0.93	454

The InceptionV3 model performs the best on healthy class, as the F1 value of the class is 0.95. The model got the lowest F1 score of 0.89 on the blast class. The weighted average of the model is 0.93. The confusion matrix :

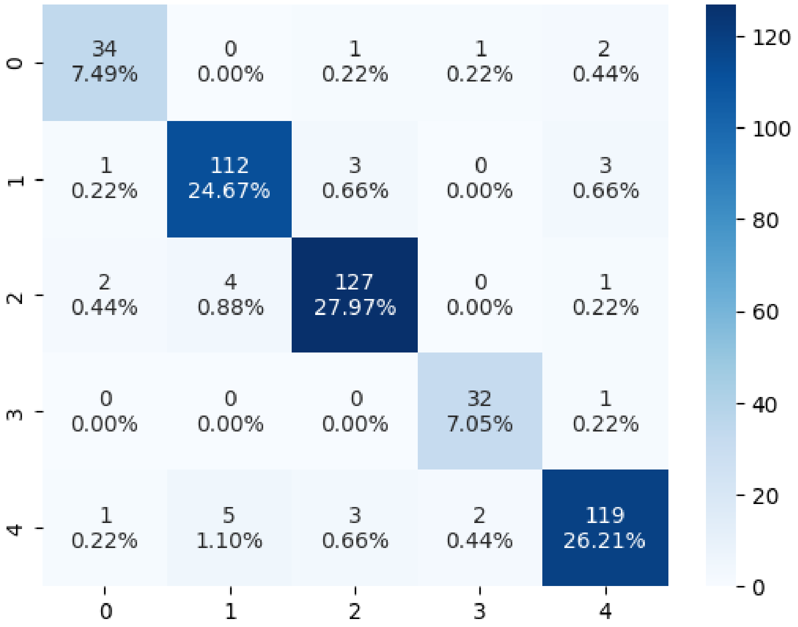


Figure 5.3: Confusion Matrix of InceptionV3

The confusion matrix 5.3 shows how good InceptionV3 has performed on the dataset, as all the cells in diagonal order are brighter compared to others.

5.2.4 ViT

Table 5.6: Classification report of ViT

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	1.00	0.92	0.96	38
Wheat Brown Rust	1.00	1.00	1.00	119
Wheat Healthy	0.96	1.00	0.98	134
Wheat Septoria	0.97	1.00	0.99	33
Wheat Yellow Rust	0.99	0.97	0.98	130
Macro avg	0.99	0.98	0.98	454
Weighted avg	0.98	0.98	0.98	454
Accuracy			0.98	454

ViT performs the best compared to other CNN models and with an F1 score of 1.00, it classifies the images of brown rust the best. ViT performs worse on blast images compared to other classes, as the F1 score for the class is 0.96. Again, the weighted average is 0.98. The confusion matrix of ViT:

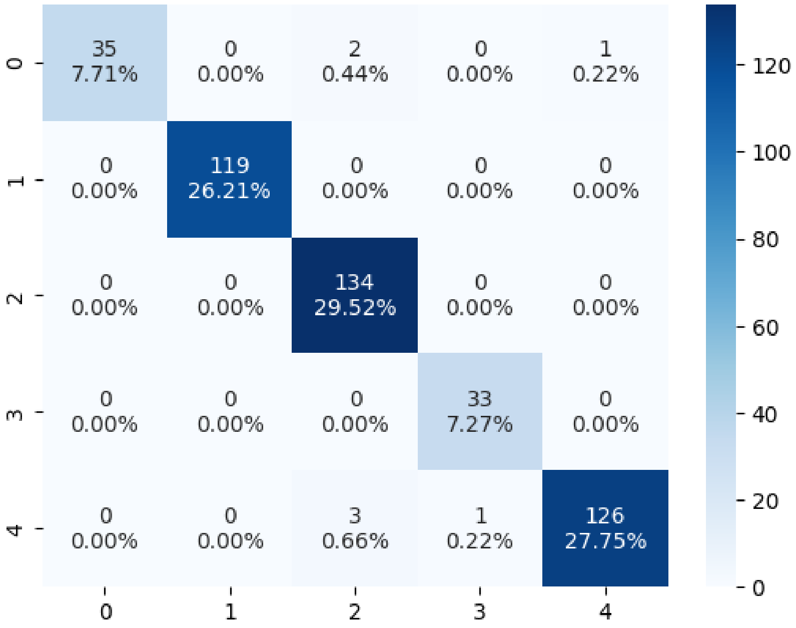


Figure 5.4: Confusion Matrix of ViT

From the confusion matrix 5.4, it can be seen that, out of 134 healthy images, ViT classified all of them correctly. However, some other classes have been misclassified as healthy images, making the F1 score of the brown rust the highest.

5.2.5 Hybrid CNN

Table 5.7: Classification report of CNN

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.97	0.97	0.97	38
Wheat Brown Rust	0.95	0.99	0.97	119
Wheat Healthy	0.89	0.99	0.94	134
Wheat Septoria	0.97	1.00	0.99	33
Wheat Yellow Rust	1.00	0.84	0.91	130
Macro avg	0.96	0.96	0.96	454
Weighted avg	0.95	0.95	0.95	454
Accuracy			0.95	454

The hybrid CNN model performs the best on the septoria class, as the F1 score on this class by the model is 0.99. In addition, the model, scores 0.97 on both blast and brown rust. Yellow rust got the least F1 score which is 0.91. However, it is better compared to that of ResNet50 and EfficientNetB0. The weighted average score of the model is 0.95. The hybrid CNN's confusion matrix:

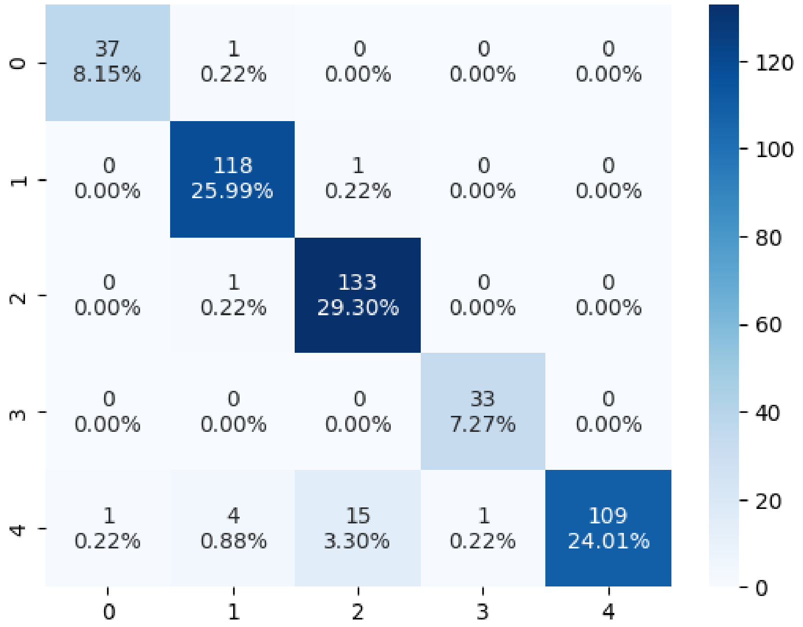


Figure 5.5: Confusion Matrix of hybrid CNN

The confusion matrix 5.5 shows that the model is classifying 37 out of 38 images of wheat blast correctly, which is pretty good. Again, the model performs good on brown rust, healthy and septoria classes as well. However, in case of the yellow rust class, the model classifies 109 images correctly out of 130 images.

5.2.6 ResNet50 & ViT

Table 5.8: Classification report of ResNet50 & ViT

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	1.00	0.95	0.97	38
Wheat Brown Rust	1.00	1.00	1.00	119
Wheat Healthy	0.97	1.00	0.99	134
Wheat Septoria	0.94	1.00	0.97	33
Wheat Yellow Rust	0.99	0.96	0.98	130
Macro avg	0.98	0.98	0.98	454
Weighted avg	0.99	0.98	0.98	454
Accuracy			0.98	454

The average or voting ensemble model of ResNet50 and ViT, performs the best on brown rust with a F1 score of 1.00. The weighted average is 0.98. The confusion matrix:

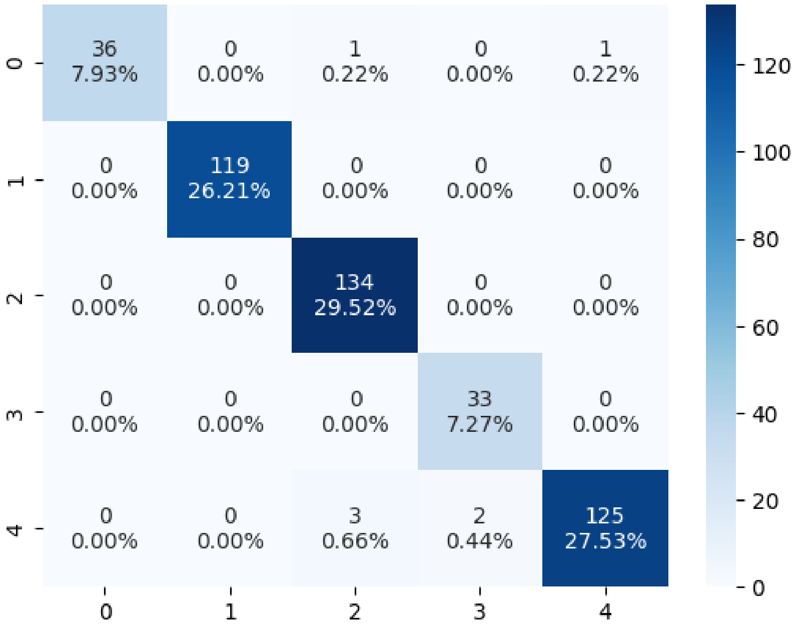


Figure 5.6: Confusion Matrix of ResnetNet50 & ViT

The confusion matrix 5.6 shows that the ensemble model classifies 35 images as septoria, although 33 images are actually of septoria. Thus, the F1 score is 0.97.

5.2.7 EfficientNetB0 & ViT

Table 5.9: Classification report of EfficientNetB0 & ViT

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	1.00	0.92	0.96	38
Wheat Brown Rust	1.00	1.00	1.00	119
Wheat Healthy	0.96	1.00	0.98	134
Wheat Septoria	0.97	1.00	0.99	33
Wheat Yellow Rust	0.99	0.97	0.98	130
Macro avg	0.99	0.98	0.98	454
Weighted avg	0.98	0.98	0.98	454
Accuracy			0.98	454

The classification report 5.9 shows that EfficientNetB0 and ViT have performed the best on the images of brown rust with an F1 score of 1.00, whereas, the ensemble model got an F1 score of 0.96 on the images of blast. The confusion matrix for the model:

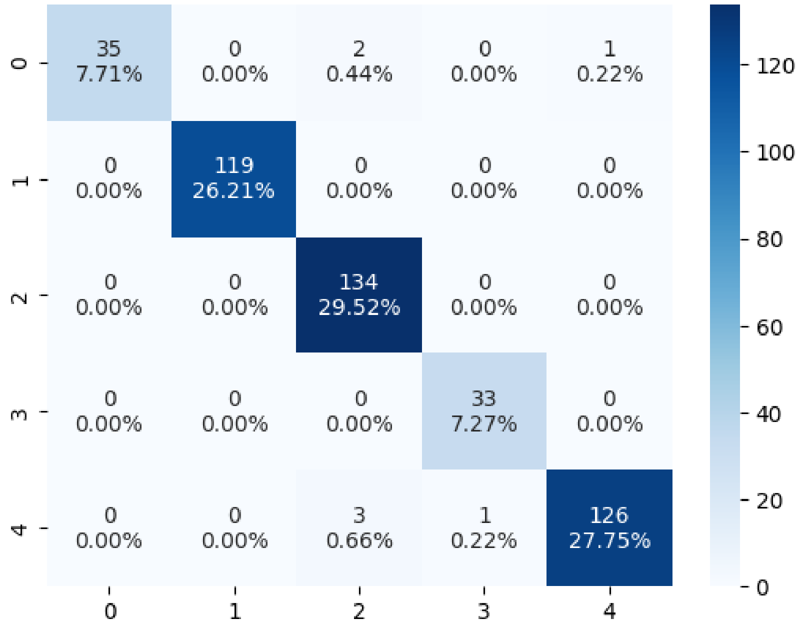


Figure 5.7: Confusion Matrix of EfficientNetB0 & ViT

The confusion matrix 5.7 demonstrates that the ensemble model of EfficientNetB0 and ViT classified 35 out of 38 images as blast, 2 as healthy and 1 as yellow rust.

5.2.8 InceptionV3 & ViT

Table 5.10: Classification report of InceptionV3 & ViT

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.97	0.95	0.96	38
Wheat Brown Rust	0.99	1.00	1.00	119
Wheat Healthy	0.99	1.00	1.00	134
Wheat Septoria	1.00	1.00	1.00	33
Wheat Yellow Rust	0.98	0.98	0.98	130
Macro avg	0.99	0.98	0.99	454
Weighted avg	0.99	0.99	0.99	454
Accuracy			0.99	454

The ensemble model of InceptionV3 and ViT gets 1.00 as the F1 score for brown rust, healthy and septoria class. The lowest F1 score it got is 0.96 for the blast class. The weighted average of the ensemble model is 0.99. The confusion matrix for the model:

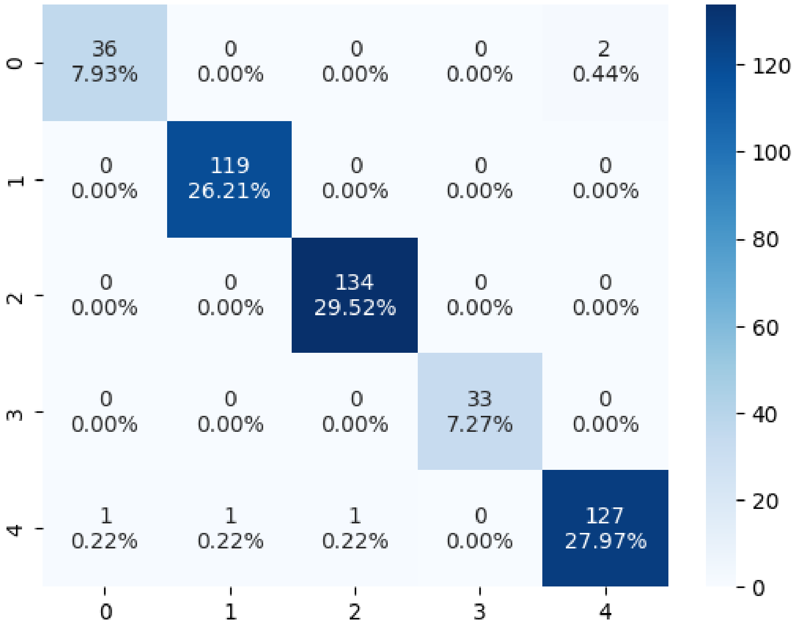


Figure 5.8: Confusion Matrix of InceptionV3 & ViT

The confusion matrix 5.8 demonstrates that InceptionV3 and ViT performed really well on the dataset, as all the cells in diagonal order are brighter than others and the number of misclassifications is less compared to that of other models.

5.2.9 Proposed Model

Table 5.11: Classification report of the proposed Model

	Precision	**Recall**	**F1 Score**	**Support**
Wheat Blast	0.97	0.97	0.97	38
Wheat Brown Rust	0.99	1.00	1.00	119
Wheat Healthy	1.00	1.00	1.00	134
Wheat Septoria	1.00	1.00	1.00	33
Wheat Yellow Rust	0.99	0.98	0.99	130
Macro avg	0.99	0.99	0.99	454
Weighted avg	0.99	0.99	0.99	454
Accuracy			0.99	454

The proposed ensemble model got an F1 score of 1.00 on brown rust, healthy and septoria which is as good as that of the ensemble model of InceptionV3 and ViT. On top of that, the proposed model got F1 score of 0.97 and 0.99 on blast and rust respectively, which are better than those of all the models. The confusion matrix of the model:

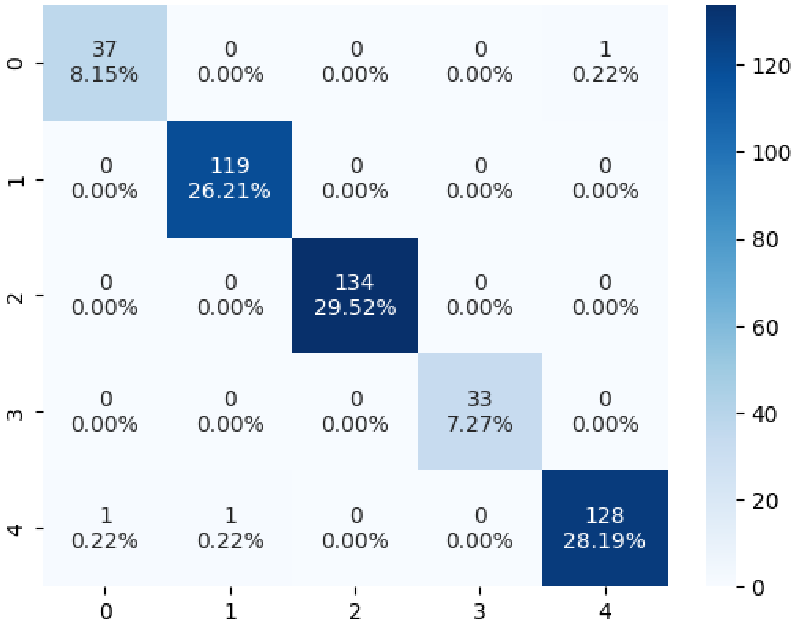


Figure 5.9: Confusion Matrix of Hybrid CNN & ViT

From the confusion matrix 5.9 it is understood that the model effectively classifies 37 out of 38 blast images as blast and only 1 as yellow rust. It performs great on the images of brown rust, healthy and septoria as there are no misclassifications for these 3 classes, resulting in an F1 score of 1.00. On the other hand, for the yellow rust class, 128 out of 130 were correctly classified as 1 was misclassified as blast and 1 was misclassified as brown rust. However, this is better compared to the other models.

5.2.10 Comparison among models

Table 5.12: Comparision of all models

Model Name	Accuracy	Epochs	Train Acc	Validation Acc	F1 Score
ResNet50	72.03%	50	0.7052	0.6816	0.72
EfficientNetB0	39.65%	50	0.3869	0.3657	0.40
InceptionV3	93.39%	50	1.0000	0.9559	0.93
ViT	98.46%	50	0.9958	0.9829	0.98
Hybrid-CNN	94.71%	50	0.9853	0.9461	0.95
ResNet50 & ViT	98.46%				0.98
EfficientNetB0 & ViT	98.45%				0.98
InceptionV3 & ViT	98.89%				0.99
Proposed Model	99.34%				0.99

From the table 5.12 it can be seen that, all the pretrained models and the hybrid CNN have been trained for 50 epochs and among the individual models the ViT tops the rank with an accuracy of 98.46%. Whereas, the training accuracy of InceptionV3 is the best, as the score is 1.00. Validation accuracy represents how well the model performed on a separate validation dataset, where ViT got highest with the value of 98.29%. Furthermore, the hybrid CNN with an accuracy of 94.71%. The accuracies of the ensemble models are: ResNet50 and ViT having 98.46%, EfficientNetB0 and ViT having 98.45%, InceptionV3 and ViT having 98.89%, whereas the proposed model having the highest accuracy of 99.34%. Lastly, considering the value of F1 score of the combined model InceptionV3 and ViT, and the proposed model is the highest which is 99%.

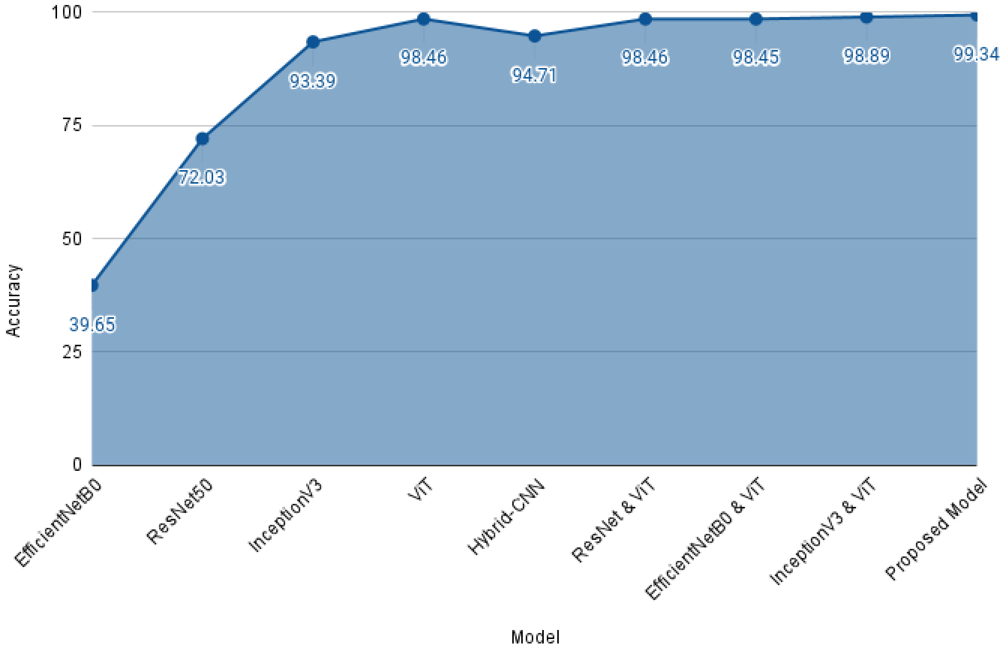


Figure 5.10: Representation of all the mentioned model's accuracy

Accuracy vs Model representation

The hybrid CNN model is at the second with an accuracy score of 94.71% followed by that of Inceptionv3 which is 93.39%. Among the ensemble models, ResNet50 and ViT perform similarly to EfficientNetB0 and ViT as these models score 98.46% and 98.45% respectively in terms of accuracy. InceptionV3 and ViT is at the second spot with an accuracy of 98.89% while the hybrid CNN and ViT perform the best among all the models with an accuracy score of 99.34%.

5.2.11 Discussion

The aforementioned analysis have shown that the individual models were tested separately. From the accuracies of these individual models, it can be seen that the EfficientNetB0 performed very poorly on the custom dataset, which might be because of the model being mostly focused on the low level features and the less number of training images. Since, the custom dataset has less number of training images, the model couldn't be trained with enough information. Thus, the effect of this issue is visible on the performance of the model. In addition, the limited size of the dataset may not provide a sufficiently varied range of samples to justify the use of a complex model like EfficientNetB0.

On the other hand, the performance of ResNet50 falls behind, which might be due to insufficient complexity, limited diversity and over-fitting etc. Another reason for poor performances might be that, small datasets are often more susceptible to noise or class imbalance issues, which can negatively impact the model's performance.

The other three models; InceptionV3, ViT and the hybrid CNN performed far better than the previous two models here. As the ViT is a Transformer model, it's the state-of-the-art model now. Other than the ViT model, among all the deep convolutional models, the hybrid CNN performed the best here. Since, this model has been built from scratch by training on the custom wheat dataset, it performed the best among all the other CNN models. Furthermore, after building the ensemble models with ViT, if we look at the performance comparison table, it is shown there that the ensemble of EfficientNetB0 and ViT did exceptionally well on the dataset. It might be due to both of the models being good at different things, which actually complements each other's strengths. As a result, the overall accuracy has been increased.

After thoroughly evaluating all the results, it is proved that the ensemble of the hybrid CNN and ViT performs better than any other models that have been tested on the custom wheat disease dataset with an F1 score of 0.99. In the proposed model, the hybrid CNN is lightweight and less complex compared to other models with 1,726,085 parameters only.

On the other hand, ViT has the highest number of parameters among the pre-trained models. However, transfer learning has been used here and thus, only 110,725 parameters are being trained on the dataset. This makes the ensemble model of the hybrid-CNN and ViT a suitable model for not only accuracy but also for faster training.

The proposed model performs the best on this dataset that is comparatively smaller than the most. However, it is hoped to achieve a good performance on a large dataset as well. Some drawbacks might be less variety of diseases in the dataset.

5.2.12 Future Improvements

Data augmentation has not been done on the dataset. However, data augmentation on the whole dataset might improve the accuracy of the proposed model, making it more generalized towards data. Again, more variety of wheat diseases could be added to the dataset, which would improve the quality of the models. Consequently, diseases having same symptoms e.g. Fusarium head blight (FHB) and wheat blast might be a significant contribution to the wheat cultivation as it is difficult to differentiate between the two disease even in labs and depending on the disease the cure will be different. Thus, images of FHB could be added to the custom dataset in the future. Lastly, a mobile application could be built so that farmers could easily detect diseases from their farm.

Chapter 6

Conclusion

Most of the people in our country depend on agriculture. Early detection of plant diseases using modern technology is of paramount importance. Relying solely on plant pathologists or botanists for detection can be time-consuming. Unfortunately, during this period, the disease may have already spread extensively throughout the field, potentially resulting in significant crop loss. Therefore, leveraging advanced technology for timely diagnosis is imperative to safeguard agricultural yields. Through this research, the hybrid CNN and pre-trained ViT ensemble model has been implemented which outperforms all the other models. Therefore, the proposed ensemble is the best model as it has the highest accuracy of 99.34% and other qualities being lightweight, having less complexity, lower parameters in detecting various kinds of wheat diseases. Developing this automated wheat disease detection system with high accuracy and low resource requirements holds the potential to yield substantial time and cost savings. By the well-organized process of identifying and diagnosing diseases in plants, this technology could significantly enhance agricultural efficiency and productivity, as timely intervention can mitigate the spread of diseases throughout a crop field, ultimately preventing extensive crop loss. Moreover, it would alleviate the need for extensive manual labor and expert intervention, allowing for more timely responses to potential outbreaks. This, in turn, could lead to more effective disease management strategies and ultimately contribute to the preservation of crop yields and economic stability in agriculture. In comparison to existing pre-trained models, we anticipate gaining greater illness detection accuracy with our suggested ensemble model.

Bibliography

[1] A. Hossain and J. A. Teixeira da Silva, "Wheat production in Bangladesh: its future in the light of global warming," *AoB PLANTS*, vol. 5, Jan. 2013, pls042, ISSN: 2041-2851. DOI: 10.1093/aobpla/pls042. eprint: https://academic.oup.com/aobpla/article-pdf/doi/10.1093/aobpla/pls042/373376/pls042.pdf. [Online]. Available: https://doi.org/10.1093/aobpla/pls042.

[2] X. Niu, M. Wang, X. Chen, S. Guo, H. Zhang, and D. He, "Image segmentation algorithm for disease detection of wheat leaves," in *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, IEEE, 2014, pp. 270–273.

[3] J. G. Barbedo, C. S. Tibola, and J. M. Fernandes, "Detecting fusarium head blight in wheat kernels using hyperspectral imaging," *Biosystems Engineering*, vol. 131, pp. 65–76, 2015, ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2015.01.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511015000136.

[4] D. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," Nov. 2015.

[5] L. Ravikanth, C. B. Singh, D. S. Jayas, and N. D. White, "Classification of contaminants from wheat using near-infrared hyperspectral imaging," *Biosystems Engineering*, vol. 135, pp. 73–86, 2015, ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2015.04.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511015000707.

[6] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, 2016, ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2016.01.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511015302476.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016, ISSN: 1664-462X. DOI: 10.3389/fpls.2016.01419. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419.

[8] V. P. Gaikwad and V. Musande, "Wheat disease detection using image processing," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 2017, pp. 110–112. DOI: 10.1109/ICISIM.2017.8122158.

[9] M. Kumar, T. Hazra, and S. S. Tripathy, "Wheat leaf disease detection using image processing," *Int J Latest Technol Eng Manag Appl Sci*, vol. 6, no. 4, pp. 73–76, 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[11] A. Hussain, M. Ahmad, I. A. Mughal, and H. Ali, "Automatic disease detection in wheat crop using convolution neural network," 2018. DOI: 10.13140/RG.2.2.14191.46244. [Online]. Available: http://rgdoi.net/10.13140/RG.2.2.14191.46244.

[12] X. Jin, L. Jie, S. Wang, H. J. Qi, and S. W. Li, "Classifying wheat hyperspectral pixels of healthy heads and fusarium head blight disease using a deep neural network in the wild field," *Remote Sensing*, vol. 10, no. 3, 2018, ISSN: 2072-4292. DOI: 10.3390/rs10030395. [Online]. Available: https://www.mdpi.com/2072-4292/10/3/395.

[13] K. A. Mottaleb, D. B. Rahut, G. Kruseman, and O. Erenstein, *Wheat production and consumption dynamics in an asian rice economy: The bangladesh case*, en, Apr. 2018. DOI: 10.1057/s41287-017-0096-1. [Online]. Available: http://dx.doi.org/10.1057/s41287-017-0096-1.

[14] J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering*, vol. 180, pp. 96–107, 2019, ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2019.02.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511018307797.

[15] M. T. Islam, K.-H. Kim, and J. Choi, *Wheat blast in bangladesh: The current situation and future impacts*, en, Feb. 2019. DOI: 10.5423/ppj.rw.08.2018.0168. [Online]. Available: http://dx.doi.org/10.5423/PPJ.RW.08.2018.0168.

[16] 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 Sensing*, vol. 11, no. 22, 2019, ISSN: 2072-4292. DOI: 10.3390/rs11222658. [Online]. Available: https://www.mdpi.com/2072-4292/11/22/2658.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[18] M. Chohan*, A. Khan, R. Chohan, S. H. Katpar, and M. S. Mahar, *Plant disease detection using deep learning*, May 2020. DOI: 10.35940/ijrte.a2139.059120. [Online]. Available: http://dx.doi.org/10.35940/ijrte.A2139.059120.

[19] H. Ma, W. Huang, Y. Jing, *et al.*, "Identification of fusarium head blight in winter wheat ears using continuous wavelet analysis," *Sensors*, vol. 20, no. 1, 2020, ISSN: 1424-8220. DOI: 10.3390/s20010020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/20.

[20] D.-Y. Zhang, G. Chen, X. Yin, *et al.*, "Integrating spectral and image data to detect fusarium head blight of wheat," *Computers and Electronics in Agriculture*, vol. 175, p. 105 588, 2020, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105588. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920306943.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[22] M. A. Genaev, E. S. Skolotneva, E. I. Gultyaeva, E. A. Orlova, N. P. Bechtold, and D. A. Afonnikov, "Image-based wheat fungi diseases identification by deep learning," *Plants*, vol. 10, no. 8, 2021, ISSN: 2223-7747. DOI: 10.3390/plants10081500. [Online]. Available: https://www.mdpi.com/2223-7747/10/8/1500.

[23] M. A. Genaev, E. S. Skolotneva, E. I. Gultyaeva, E. A. Orlova, N. P. Bechtold, and D. A. Afonnikov, "Image-based wheat fungi diseases identification by deep learning," *Plants*, vol. 10, no. 8, 2021, ISSN: 2223-7747. DOI: 10.3390/plants10081500. [Online]. Available: https://www.mdpi.com/2223-7747/10/8/1500.

[24] L. Goyal, C. M. Sharma, A. Singh, and P. K. Singh, "Leaf and spike wheat disease detection classification using an improved deep convolutional architecture," *Informatics in Medicine Unlocked*, vol. 25, p. 100 642, 2021, ISSN: 2352-9148. DOI: https://doi.org/10.1016/j.imu.2021.100642. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914821001313.

[25] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Computers and Electronics in Agriculture*, vol. 186, p. 106 184, 2021, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2021.106184. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921002015.

[26] Z. Saeed, A. Raza, A. Qureshi, and M. H. Yousaf, "A multi-crop disease detection and classification approach using cnn," Oct. 2021, pp. 1–6. DOI: 10.1109/ICRAI54018.2021.9651409.

[27] R. C. Bernardes, A. De Medeiros, L. da Silva, *et al.*, "Deep-learning approach for fusarium head blight detection in wheat seeds using low-cost imaging technology," *Agriculture*, vol. 12, no. 11, 2022, ISSN: 2077-0472. DOI: 10.3390/agriculture12111801. [Online]. Available: https://www.mdpi.com/2077-0472/12/11/1801.

[28] M. Hossen, M. Mohibullah, C. Muzammel, T. Ahmed, and S. Acharjee, "Wheat diseases detection and classification using convolutional neural network (cnn)," *International Journal of Advanced Computer Science and Applications*, vol. 13, Jan. 2022. DOI: 10.14569/IJACSA.2022.0131183.

[29] L. Li, Y. Dong, Y. Xiao, L. Liu, X. Zhao, and W. Huang, "Combining disease mechanism and machine learning to predict wheat fusarium head blight," *Remote Sensing*, vol. 14, no. 12, 2022, ISSN: 2072-4292. DOI: 10.3390/rs14122732. [Online]. Available: https://www.mdpi.com/2072-4292/14/12/2732.

[30] B. Polavarapu and H. Mamidipaka, "Blur image detection and classification using resnet-50," *i-manager's Journal on Image Processing*, vol. 9, no. 2, p. 37, 2022.

[31] Outlook, *Explainer: How did russia-ukraine war trigger a food crisis?* [Online]. Available: https://www.outlookindia.com/business/explainer-how-did-russia-ukraine-war-trigger-a-food-crisis--news-203160.

[32] M. Suman, *Wheat imports rise 116pc in six years.* [Online]. Available: https://www.thedailystar.net/business/news/wheat-imports-rise-116pc-six-years-1968185.

[33] A. N. T, *Inception v3 model architecture.* [Online]. Available: https://iq.opengenus.org/inception-v3-model-architecture/ (visited on 2021).