International Journal of Science & Technology (PRAGYANAM)
Volume 1 • Issue 1 • January–December 2025
ISSN:

A Learning based Model for Wheat Disease Detection and Classification

Rakesh Garg[1], Neetu Singla[2], Charu[3], Sachin Lalar[4] & Shubham Goel[5]

ABSTRACT

Wheat is an important staple crop for large sections of the population for many cultures across the globe. It is vital that one of the major impediments to its cultivation, namely disease, be prevented and controlled appropriately. A possible strategy for controlling yield loss due to disease is early identification to prevent their spread and minimize losses. An automated method must be developed to identify disease and that can be done through the implementation of computer vision. Such a tool can be used in drones on a larger scale or possibly smart phones on a smaller scale to identify diseases in wheat. A computationally efficient model for early detection of wheat crop diseases using a pre-trained convolutional neural network is developed in this research and is compared with models as VGG16, MobileNet, and ResNet50, demonstrating superior performance in terms of computational cost and accuracy, precision, recall, and F1-score.

Keywords: Wheat Disease Detection, Deep Learning, Image Classification, Convolutional Neural Networks

INTRODUCTION

Wheat is one of the most popular cereal grains consumed worldwide. Experts estimate its cultivation began around 10,000 years ago in southeast Turkey. The most popular variety of wheat is bread wheat, sometimes known as common wheat. Compared to other food crops, it is cultivated on the most acreage. (220.4 million hectares, 2014). More wheat is traded globally than all other crops combined. With global productions of 764 million tons of wheat in 2019, and 772 million tons are anticipated to be produced in 2020. It is a good source of carbohydrates, as well as several vitamins and minerals. Particularly whole wheat provides a lot of advantages for our health. Different types of wheat are commonly used in everyday food products. For example, Club wheat (T. compactum) is a softer variety used in making cakes, crackers, cookies, pastries, and flours. Durum wheat (T. durum) is primarily used for producing pasta varieties such as spaghetti and macaroni, while common wheat (Triticum aestivum) is typically used in bread-making. Diseases account about 20% of the annual loss of wheat. 50 of the 200 wheat illnesses generate monetary losses and are widespread. The proposed approach achieved impressive results, with a recognition rate of up to 91%, by utilizing an ANN classifier for classification and a Gabor filter for feature extraction. The ANN classifier combines various features, including textures, colors, and patterns, to effectively categorize and identify different plant diseases. Around the world, fungus-related wheat illnesses result in production losses of roughly 20% and have diverse effects on grain quality. New pathogen races emerge often, well-known illnesses impact new hosts, and newly developing diseases pose a danger to wheat production. A large range of fungal pathogenic diseases pose the greatest danger to wheat because they cause significant crop loss. Rusts, smuts, Fusarium head blight, Septoria leaf blotch, tan spot, and powdery mildew is among the pests that cause the most damage. It is predicted that grain yields must increase by at least 70% by 2050 to fulfil the demands of a worldwide population rising at the current rates, given the pre-existing global hunger issue. It is not feasible to limitlessly expand land under

[1] Computer Science & Engineering Gurugram University, Gurugram. E-mail: drrakeshgarg@gurugramuniversity.ac.in
[2] Computer Science & Engineering Gurugram University, Gurugram. E-mail: neetusingla@gurugramuniversity.ac.in
[3] Computer Science & Engineering Gurugram University, Gurugram. E-mail: drrcharu@gurugramuniversity.ac.in
[4] Computer Science & Engineering Gurugram University, Gurugram. E-mail: sachinlalar@gurugramuniversity.ac.in
[5] Computer Science & Engineering Gurugram University, Gurugram. E-mail: shubhamgoel@gurugramuniversity.ac.in

architecture because the amount of land that can be used for agriculture is intrinsically limited—whether by fertility, water availability, or owing to the inherent restrictions on land. Therefore, we need to figure out how to increase yields in the current agricultural area. The prevalence of disease in crops is a significant barrier to achieving this goal. According to studies, worldwide grain production losses from diseased crops range between 18% and 21.5%, and in individual wheat hotspots, they range between 10.1% and 28.1%. Given the serious effects of wheat diseases and the importance of wheat as a commodity in the world's food markets, it is crucial that they are identified early and treated.

Disease detection in particular plants presents a problem because it is difficult and expensive to collect large, labelled datasets for them due to a variety of factors, including local weather conditions and challenges in locating experts in the field to identify the diseased plants for which the dataset can be collected. Large datasets for these challenges are therefore difficult to construct or, in certain situations, just not practical. Consequently, in these situations, the datasets are somewhat small (less than 5000 images). When training deep CNN models from scratch, smaller datasets need more time and resources, such as computing hardware (GPUs, CPUs, and RAM), because these models must be trained with a relatively high number of parameters, which increases their computational complexity. Additionally, used smaller datasets mean fewer samples for the model, which can result in overfitting the model. In other words, the model will perform well on training data, but will have limitations with testing and new data because there isn't enough observed variance in the data for the model to be able to generalise the data effectively to use with new data points. Transfer learning, where we leverage models pretrained on a separate, large dataset, can be used to solve deep learning problems with small datasets. Instead of utilising random weights while training the model on the real dataset, the weights learned through pretraining might be employed. By using pretrained weights to reduce the number of parameters to be trained, by "fixing" some of the initial layers, i.e., using their parameters as-is, and only training the final few, the model can apply the knowledge gained from pretraining dataset without having to start from scratch. As a result of the decreased number of parameters that must be learned the training time and computational needs of the model are both greatly reduced. Furthermore, the model only has to be significantly modified to the specific situation because it already has enough data to make generalisations from the pretraining data. Transfer learning therefore offers a practical, affordable way to train a deep learning model for issues where data is either unavailable or difficult to gather. Disease detection technologies based on mobile applications is a viable use case for disease detection in wheat to improve grain yields. Given that over 90% of people worldwide own a mobile phone, farmers have the opportunity to utilise a smartphone app to identify illnesses in their crops and request assistance as needed. Without specialised assistance, we must rely on visual disease diagnosis in wheat. Since the most common illnesses in wheat tend to be fungal infections that emerge in the leaves, this may be accomplished by categorising photographs of wheat leaves. Convolutional neural networks (CNNs) are one of the most crucial techniques for classifying issues in pictures [1]. In order to solve the problem of disease detection and classification in wheat, a CNN architecture has been proposed in this paper, which draws inspiration from the already existing models and optimises the same in terms of the computational requirements, further leading to the reduction in time taken to evaluate the given input, offering viable real- world application of the model on less powerful machines like more affordable smartphones.

The rest of this paper is organised as follows. Section II reviews the existing literature on illness detection approaches such as convolutional neural networks. Section III describes the architecture of the proposed model, highlighting its different elements. Section IV present the stepwise implementation of proposed model for wheat disease detection, performance evaluation of proposed model followed by comparison with state of art models. Section V provides the major findings of this research followed by the conclusion and future scope in the last section.

LITERATURE REVIEW

There exists a history of literature solving disease classification problems in plants, including maize, potatoes, rice, and wheat among others [2]. Many authors have proposed solutions to this problem in the form of various architectures. Most of these employ CNNs to perform image classification, with some using self-created CNN architectures and some adapting popular architectures for solving the same[3]. In [4], the authors assessed the performance of support vector machines (SVM), backpropagation neural networks, and generalized regression neural networks, comparing them with traditional multiple regression methods. The study found that an SVM-based regression approach provided a more effective representation of the relationship between environmental factors and disease severity, offering potential benefits for disease management. Further, image processing techniques [5] have been employed for the automated classification of plant diseases through the analysis of leaf images. The study employs the SVM classifier to differentiate between healthy and diseased soybean leaves. To assess the system's performance, a dataset consisting of 120 images, captured from multiple farms using different mobile cameras, was used. The SIFT algorithm enables precise identification of plant species by analyzing leaf shapes. With an average accuracy of 93.79%, the SVM classifier effectively distinguishes between healthy and sick soybean leaves. Researchers in [6] used deep learning technologies to identify diseases in various plants. One of the key processes in their methodology was the in-depth image processing, including histogram equalization, noise filtering, and decolorization, and various image segmentation techniques. By separating the image into different parts and studying each section separately, image segmentation helps to make image identification and analysis simpler. All parts share the same qualities in terms of color, texture, and intensity. The segmentation is a region-based technique to differentiate the unhealthy and healthy parts of the leaves on the bases of color. This approach performed well with multiple plants and crops, highlighting the importance of in-depth image processing techniques. The HOG approach for feature extraction has been applied [7], where, a histogram of gradient orientation representation over the pictures is computed using the HOG, which is utilised to partition the image into distinct portions. By counting the incidence of the gradient orientation, it collects features. HOG is required in many sectors of object identification, including as face recognition and in our study, for plant leaf recognition. Application of Artificial Neural Network technology and a variety of image processing techniques is done to provide a methodology for early and precise plant disease identification [8]. The suggested method produced superior results with a recognition rate of up to 91% since it is based on an ANN classifier for classification and a Gabor filter for feature extraction. An ANN-based classifier employs a mix of textures, colors, and characteristics to categories various plant diseases and identify them. Authors in [9] used AlexNet architecture to classify wheat disease, with accuracy of 84.54%. They employed the AlexNet architecture to train their model but owing to a smaller dataset chose to pretrain the model on other data to initialize weights. All the photos in the dataset were downsized to '227 × 227' in accordance with the requirements of the AlexNet architecture. By using ReLu and MaxPooling, features were retrieved from the convolutional layers of the CNN model. Using powerful machines, the authors trained the deep architectures on a sizable dataset like ImageNet in the pre training phase. It was inferred from the paper that the smaller dataset would lead to overfitting problem. Hence the concept of using a model which has already been trained on a larger dataset was developed so that the model generalizes better. Alternative method is to train only the final classification layer while freezing the weights of the first layer. Further a method for classifying and diagnosing four common diseases affecting apple leaves has been proposed in the study put out by [10], which in comparison to a typical AlexNet model, the architecture achieved an overall accuracy of 97.62% with significantly fewer parameters. It has been observed that advancements in CNN performance are primarily driven by the development of new blocks and structural redesigns [11].

A variation of LeNet architecture [12]. has an additional block of convolutional, activation, and pooling layers. In this architecture, three such blocks are used, followed by completely connected layers and softmax activation. While fully connected layers are used for classification, convolutional and pooling layers were utilised to extract features. The max-pooling layer was used to speed up training and reduce the model's sensitivity to small changes in input by reducing the size of the feature maps. Each of the blocks employed the ReLU activation layer to introduce nonlinearity. In order to prevent overfitting, the train set, the Dropout regularisation algorithm had also been applied with a probability of 0.4. Dropout regularisation reduces model variance and streamlines the network, which helped minimise overfitting by randomly removing neurons from the network after each iteration of training. LeNet was put forward as an explainable model that could reliably, promptly, and precisely identify and measure leaf tension [13], which further aimed at developing a 95.04% accurate real-time detection system for mobile platforms in order to quickly and broadly observe crips in actual production conditions. A CNN architecture known as the supervised 3D-CNN based model was introduced [14] to learn the spectral and spatial information of hyperspectral images for the categorization of healthy and samples that have been exposed to charcoal rot. The hyperspectral wavelengths that significantly improve classification accuracy were found using a saliency map-based visualization technique. The model's categorization accuracy was 95.73%. Authors in [15] illustrated the extraction and categorization of groundnut leaf disease using color imagery. With the help of a neural network, the color imaginary transform, color co- occurrence matrix, and feature extraction were carried out. With a complicated backdrop, back propagation proved to be effective in groundnut leaf detection, and was successfully able to diagnose the illnesses.

A CNN, that made use of data-augmentation, transfer learning, and MBGD as an optimizer, has been used [16] to obtain an overall testing accuracy of 97.61% with a loss value of 0.35. Loss functions were applied to assess the model's performance. Categorical Cross-Entropy is one such loss function used in this study for multi-class categorization. It contrasts the projections' distribution with the labels' actual distribution. For all other classes, the likelihood is maintained at 0, but the probability of the correct class is maintained at 1. Research presented in [17] employed a 5-layer self-designed convolutional network, along with a regression layer to identify diseased leaves. They explained that the neural network is composed of various layers, including the input layer, convolutional layer, output layer, and fully connected layer. They noted that more layers can be added using the convolutional layer. The first step is to load the input data, followed by creating the convolution layer. It is mentioned that every layer has an activation function. They discuss the combination of a pooling function with a convolutional neural network. In this particular case, they have constructed five convolutional layers and added matching pooling. They have taken the last fully linked layer and applied a softmax activation function at the end of each layer. In order to receive the result and employ the optimizer, the regression layer is used at the end. A self-designed 6-layer-deep convolutional neural network architecture is used to identify leaf disorders in multiple plants [18], building upon it with addition of other standard algorithms, such as the ADAM optimizer and the softmax classifier. The ReLu activation function, the picture input shape of '256, 256, 3', the filter size of '64', the kernel size of '88', the Padding, and the strides of '11' were all included in the first convolutional layer. The second convolutional layer exhibited the same form as the first layer and added some more features. The image size and the kernel size were reduced in the subsequent layers. They employed the ReLu activation function and max pooling layers with the ADAM optimizer, using an additional softmax layer to classify the healthy and diseased leaves, achieving an accuracy of 96.28% on testing dataset. The use of image expansion techniques like alteration of image shapes and angles have led to higher accuracy when dealing with redundant data. HOResNet [19], a different ResNet-based architecture that aims for a robust recognition of plant diseases was proposed. With photos captured in actual agricultural situations, the study investigates the issue of low precision in the diagnosis and classification of plant diseases. By experimenting with photographs of various sizes, shooting angles, positions, backgrounds, and lighting, it strengthens the ability to resist influence. The outcomes showed that the method was accurate in detecting diseases between 90.14% and 91.79%. A CNN-based approach of identifying anthracnose lesions by employing data augmentation methods using Cycle-Consistent Adversarial Network (CycleGAN) [20], used a DenseNet to enhance the low-resolution resource layers of the YOLO-V3 model, resulting in accuracy of 95.57%. A novel deep convolutional neural network (DCNN)-based method for the identification of yellow rust crop disease [21], utilises extremely high spatial resolution hyperspectral pictures taken with UAVs. The suggested model included a number of Inception-ResNet layers for feature extraction and was tuned to determine the network's ideal depth and breadth. The idea uses three-dimensional data and can identify yellow rust in wheat with an accuracy of 85% by using spatial and spectral data. OR-AC-GAN is a technique discussed in [22], that was created for the early identification of the tomato spotted wilt virus utilizing hyperspectral images and an auxiliary external removal classifier using opposed generation networks. Plant segmentation, spectrum classification, and picture classification are all included in the concept. The findings indicated that the accuracy reached 96.25% before the onset of apparent symptoms. The deep study of the available literature points towards some common limitations of the existing methods for wheat disease detection as given below.

- Computational Complexity, the hardware needs required to run these models are significant, limiting total model utilisation and even additional model advancements such as detection of new illnesses or diseases in other crops.
- Training time, or the time required to learn from the database and adjust its internal parameters for making future predictions is high and will continue to rise proportionally with the size of the dataset, making it difficult to update the underlying database on a regular basis to improve the results.
- Inconsistent findings across many performance criteria; aside from accuracy, other performance metrics were inconsistent in all circumstances.

MATERIALS AND METHODS

A deep learning-based framework for wheat disease detection and classification is developed using the convolutional neural network as shown in Fig. 1. The description of all the components of the developed model is provided below.

Convolutional Neural Network

Convolutional Neural Networks have made significant advancements over the past decade in fields pertaining to pattern recognition such as voice recognition and image processing. One of major reasons behind the desirability of Convolutional Neural Networks over Artificial Neural Networks is the reduction in the parameter count [23], thus making it a viable solution for researchers to build bigger models to solve much complex problems that could not be addressed by regular ANNs. One of the most crucial assumption of

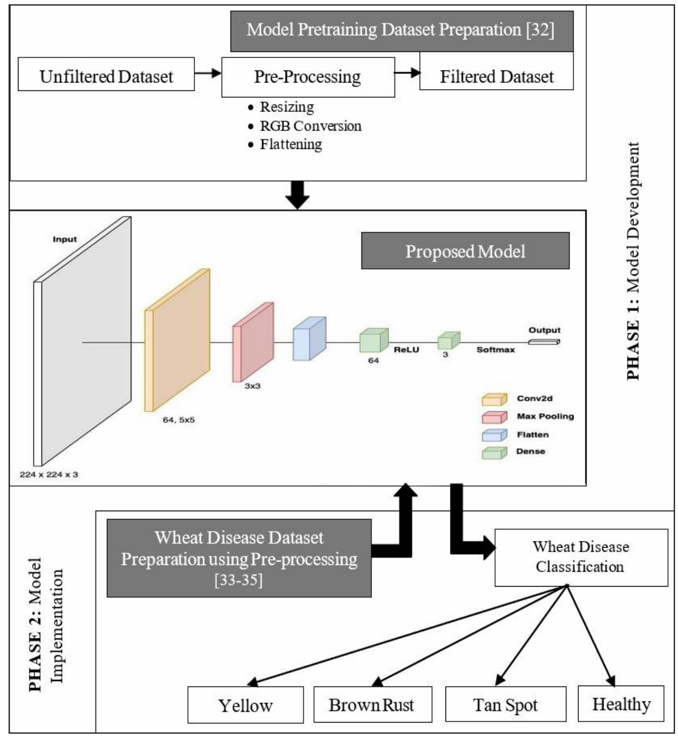

Fig. 1. Architecture diagram of proposed model for wheat disease detection and classification

problems that CNN solves is that they involve spatially independent features, in the case of a leaf detection tool, the primary focus is discovering the leaf in an image regardless of its positioning in that image. Another important aspect of CNN is its ability to extract abstract features while input propagates to deeper levels, in the example of an image detection tool, the first layer could extract the boundary or edges of the image, followed by the second layer extracting simpler forms and finally the higher-level features such as the object such as the leaf being extracted in the subsequent layers [24]. Due to the reduction in the parameter count and the ability to extract features while input propagation, as compared to other algorithms for pattern-based classifications, CNNs require substantially lower pre-processing [25]. CNN have the ability to automatically and adaptively learn filters through the process of back- propagation supported by convolution, pooling and a fully- connected layer in order to extract the relevant features for the classification, taking away from manually preparing these filters in the traditional approaches. This ability enables the network to better understand the complexity of the image, by providing a better fit for the image dataset by reduction in the number of parameters to be examined as well as the ability to reuse the weights.

Layers

- **Pooling:** Pooling is a significant step in any convolution- based networks, it helps in compressing the feature maps extracted in the previous layer. It preserves important information and removes unnecessary information by combining a group of values into a smaller group of values, thus turning the joint feature representation into valuable information. This further enables the model to be trained properly only on the dominating features that are spatially invariant, i.e. not affected by the position or the rotation of the object being classified. Polling is further divided into two classes: max pooling and average pooling. Max pooling yields the greatest/ largest value from the region covered by the Kernel on the image. Whereas, average pooling yields the average of all the values from the region covered by the Kernel on the image. Our model makes use of the Max Pooling class, it not only derives the highest values from the region but also helps in de-noising as well as completely ignoring the defective activations. Average pooling, on the other hand, only compresses the feature map as a noise suppression technique. Thus, max pooling out performs average pooling and makes it more fit for application with our proposed model.
- **Flatten and Dense (Fully Connected Layer):** Fully connected layer is (usually) an easier and cost-efficient way of learning non-linear combinations of high-level characteristics observed from the result of the convolutional layer, enabling the fully-connected layer to learn a function that may or may not be linear in that area [26]. The initial convolutional layer in a neural network has to know the dimension of the picture that is provided to it as input. The output of the picture will be provided to the dense layer after it has been processed through all convolutional layers and pooling layers. Because the convolutional layer's output is multidimensional and the dense layer's input is single- dimensional, or a 1-D array, we cannot pass the convolutional layer's output directly to the dense layer. As a result, between the convolutional and dense layers, we will utilise the Flatten() technique. A multidimensional matrix is reduced to a single dimension via the flatten() technique. In neural networks where data is processed in a single direction, the input received is flattened and back propagation is used in each of the training cycle. The model with the help of classification methods such as softmax is able to characterise images across a number of epochs by identifying dominating and specific low-level features. A dense layer is deeply connected with its preceding layers, where each of the neuron in the dense layer receives input from all of the neurons of the previous layer. The results from the convolutional layer serve as a base upon which the dense layer is applied to categorize the images [27].

Activation Function

The neurons calculate a weighted average of their input in each of the layers of the neural network, the output of the same being processed through a non-linear function called the "activation function" serves as the neuron's output. This procedure is applied throughout the neurons present in all the levels of the neural network. The convolution neural network's activation function is a crucial component. A nonlinear activation function is typically used to map the calculated features in the three phases of a convolution neural network, convolution, sub-sampling, and full-connection, in order to overcome the problem of inadequate expressiveness produced by linear operation [27].

- **ReLU:** The Rectified Linear Activation Unit, has become the norm for a variety of neural networks because of its ease to train and the frequent performance enhancements. Depending upon the input, if the input is positive the ReLU returns the input as it is for the output but if the input is negative, the ReLU returns 0 in place of the original input, thus resulting in reduction in the overall computation going forward. The Rectified Linear Activation Function provides a solution for converting a non-linear function to behave like a linear function which is a mandatory requirement for training a deep neural network using stochastic gradient descent with back propagation of mistakes. A ReLU, is a component that performs the REL function or the change in behaviour of a non-linear function to a linear function. Further, rectified networks make use of rectifier function in the hidden layers. Due to the fragility of ReLu activation neurons, some input during training may fall into the hard saturation area, leading to permanent neuronal death, preventing the updating of the appropriate weight. Additionally, the Relu function generates the output with migration phenomena by setting a portion of the neuron's output to zero. Such obnoxious forced sparse processing may hide numerous beneficial characteristics, having a negative impact on the effectiveness of model learning. Excessive sparsity may cause increased error rates and lower the model's useful capacity. The convergence of the network may be impacted by both the migration phenomena and neuronal death.

- **Softmax:** The Softmax Function, a generalisation of the Logistic Function, guarantees that our forecast adds up to 1. The Softmax function usually serves as the activation function at final layer in a neural network to normalize the output of a network to a probability distribution over predicted output class. The Softmax function and the cross-entropy function are closely related, on application of the Softmax function, the cross-entropy function is acts as the loss function in the network to validate the model's correctness as well as to improve the functioning of the network. The cross entropy helps overcome issues such as the output value being much smaller than the true value at the start of the back propagation, by slowing the gradient descent. The Softmax function reduces the vector of K real values into a vector of K real values that sum up to 1 by converting all of the varying values in the vector to be between 0 and 1, thus enabling it to be interpreted in terms of probability. It translated little or negative values into smaller probability and on the other hand, the larger or positive values into probabilities, but always summing up to '1'.

WHEAT DISEASE DETECTION AND CLASSIFICATION

This section presents the stepwise procedure of wheat disease detection and classification using the developed model described in earlier section-2. The stepwise implementation procedure for the developed model is given in algorithm-1.

Algorithm-1: Developed model for wheat disease Detection and Classification
• To begin, the actual and pretraining datasets are compiled. In order to prepare the images for learning, they are pre-processed by resizing, rescaling, and formatting the color space. The images are then flattened into vectors for processing. • Next, the compiled pretraining dataset is run through the model. This helps initialize the weights to more relevant values for training, rather than relying on random initialization of weights. • Then, the actual dataset is split into training and testing sets using random selection. • Next, the training dataset is run through the pretrained model. This step calibrates the preinitialized weights to the actual dataset for prediction on unseen data, resulting in the final, fully trained prediction model. • To evaluate the model, the testing dataset is run through the final model and assess it using various selected metrics.

Dataset Description

For the pre-training dataset, the data was compiled from the freely available [28]. A neural network is pre-trained by first applying the model to a single task or dataset. Afterwards, the model is trained on a different task or dataset using the parameters or model from previous training. By doing this, the model gains an advantage over beginning from scratch. This dataset contains approximately 87K RGB images of healthy and diseased leaves and which is divided into 38 classes that include crops of apple, bell pepper, cherry, corn, grapes, orange, peach, blueberry, potato, raspberry, soyabean, strawberry, and tomato, some of which are shown in Table 1 along with their distribution. For the actual training dataset information was gathered from a number of publicly accessible databases [29]. Images of the affected plants were gathered from various databases for the three diseases that account for the majority of the yield loss in wheat: Tan Spot, Leaf Rust, and Stripe Rust.

- P. striiformis Westend. F. sp. tritici (Pst), a pathogen found in temperate areas with chilly and damp conditions, is the root cause of wheat stripe (yellow) rust [30]. Found mostly on the leaves, but also on the stem and leaf sheaths. Pst affects over 88% of the wheat types in the globe, causing roughly US$ 1 billion in losses annually.

- Leaf rust, the most common and widespread of the wheat rust diseases, is caused by (Puccinia triticina Eriks [31-33]. Although the timing and location of grain losses brought on by leaf rust vary, the disease has a considerable economic impact [34-35]. More than US$ 350 million in damages are thought to have been incurred by Pt in the US between 2000 and 2004. Leaf rust is a difficult disease to cure due to the pathogen's great diversity, the regular appearance of novel virulence profiles, and the pathogen's strong tolerance to a wide range of temperatures.
- Tan spot, also known as yellow spot or yellow leaf blotch, is a foliar spotted disease caused by yrenophora tritici repentis that affects all major wheat-growing regions worldwide [35]. Average yield losses are 50%, although yield losses of up to 50% have been observed in disease- prone regions [36].

To prevent innate biases from showing up in the data, the distribution of the photographs was balanced as shown in Table 1, and they were carefully chosen for their quality and suitability for our intended use.

Table 2. Sample images (each class)

Class Name	*Images*	*Sample Source Image*	
Yellow Rust	924		
Brown Rust	902		
Tan Spot	910		
Healthy	1116		

Pre-processing

Real-world datasets are rarely in a usable state as-is, thus some operations need to be performed on the data to optimise it for machine learning. For image data, this can include operations like rescaling, resizing etc. as well as noise reduction, enhancement, normalisation to improve image quality for further processing. CNNs are designed to automatically learn and extract important information from raw image data, which reduces the need for feature engineering, however based on input images, pre-processing may be required to standardise input, reduce noise and format the images. For pre-processing the compiled dataset, which was assembled from several sources and had various dimensions, we had to resize and rescale images into a standard 224x224x3 format. To make them easier to utilise with the models, these were then compiled into a.csv file in the form of flattened vectors. Further for the purpose of comparison, for each of the existing model, to adapt the images to the models, the pre-processing functions offered by the Keras API was implemented, which include some fundamental pre-processing functionality. For example, in the cases of VGG16 and ResNet50, images were converted from RGB to BGR format, which is the format in which the model was pretrained and thus the pre-trained weights were stored.

Performance Metrics

The following set of classification metrics were used to compare the various aspects of each of our selected models and analyse them based on them because none of the metrics available for assessing the effectiveness of machine learning and deep learning models alone offers a complete picture of the performance of the model.

1. **Accuracy:** A measure of how well the model's predicted values match the actual values. This being a categorization issue, the accuracy is easily determined as:

$$\text{Accuracy} = \frac{\text{\#Data Points Classified Correctly}}{\text{\#Total Data Points}} \quad ...(1)$$

2. **Precision:** Precision measures the proportion of affirmative identifications that are in fact accurate. In other words, for a specific class:

$$\text{Precision} = \frac{\text{\# No. of True Positives}}{\text{\#True Positives} + \text{\#False Positives}} \quad ...(2)$$

3. **Recall:** Recall aims to quantify the percentage of true positives for a given class that are properly classified. Mathematically,

$$\text{Recall} = \frac{\text{\# No. of True Positives}}{\text{\#True Positives} + \text{\#False Negatives}} \quad ...(3)$$

4. **F1-Score:** As stated, precision and recall are counteractive measures to each other. However, they are both important to the model. They are counter to each other, in that attempting to increase precision tends to decrease recall and vice versa. To

solve this problem, the F1-score may be used [37]. The F1- score is the harmonic mean of precision and recall, and a high F1-score is a singular measure that can help us identify a model's performance, as a high F1-score means both precision and recall are significant, while a low F1-score means either or both of them are very low, which is undesirable.

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \text{...(4)}$$

5. **Number of Parameters**: The number of parameters the model must learn before it may be used to forecast data points that have not yet been observed. The number of parameters directly affects the computational efficiency of the model since the more parameters there are, the longer it will take to train the model and the longer each prediction will take to make when using it on data that has not yet been seen [38].

Training metric graphs are an essential tool for evaluating the performance of a Convolutional Neural Network (CNN) during training. These graphs typically show the improvement of accuracy and reduction of loss over training epochs, respectively. The accuracy metric measures the percentage of correctly classified examples during training. It is a measure of how well the model is able to classify input data correctly. The loss metric, on the other hand, measures the difference between the predicted output of the model and the actual output. It is a measure of how well the model is able to fit the training data. Overall, the training metric graphs are a valuable tool for evaluating the performance of a model during training and can provide insights into both the learning capability and optimization process of the model. The training metric graphs for the proposed model are presented in Fig. 2, which illustrate the evolution of accuracy and loss over the training epochs. The graphs indicate that the model achieves a high accuracy score and low loss within a few epochs of training, demonstrating that the model quickly learns to classify the input data with high accuracy and minimal error. This observation suggests that the proposed model has a strong capability to learn and generalize from the training data, which is a desirable property for any machine learning model. However, it is important to note that the performance of the model on the training data may not always reflect its performance on unseen data, and further evaluation on a separate test set is necessary to confirm the generalization ability of the model, which are discussed earlier.

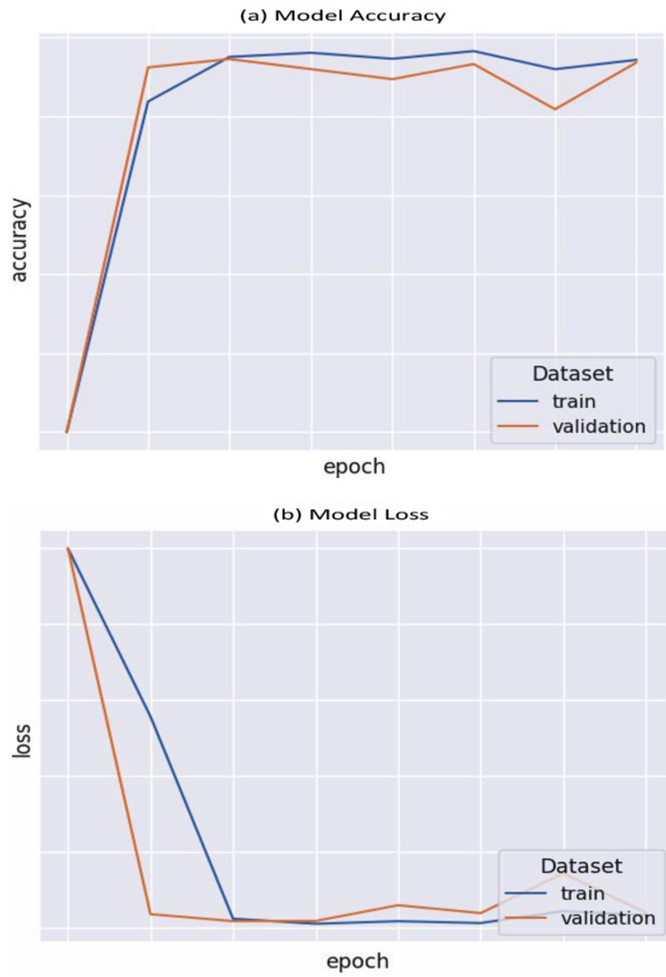

Fig. 2. (a) Model Accuracy, and (b) Model Loss

Performance Evaluation of Proposed Model

In this research, the dataset described earlier in this section, was split in the ratio of (70:30), where 70%

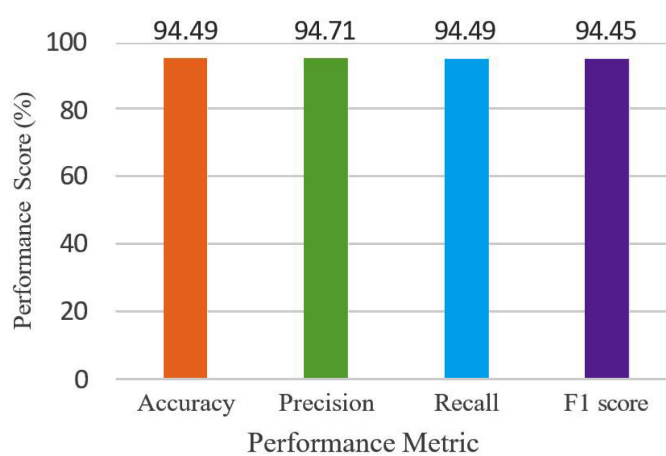

Fig. 3. Performance Statistics of Developed Model

data was used to train the developed model and 30% was used for model testing. The performance of the developed model for wheat disease detection and classification is measured on four performance metrics as shown in Fig. 3.

Comparison with State-of-Art Models

The developed model for wheat disease detection and classification is also compared with three pre-trained models as VGG16, MobileNet and ResNet50 in order to gauge its effectiveness and performance. The description about these three existing models is briefed below.

1. **VGG16:** The VGG16 model [39], along with the VGG19, was created for the 2014 ImageNet Challenge, where they won numerous problems. It is a popular deep learning architecture for image classification datasets. It includes 16 layers, including a fully connected layer for classification and a convolutional layer stack of 13 levels for feature extraction. It has 1000 output classes and a 224 × 224 × 3 picture input layer. The same input structure is used for our purposes, the fully connected layers were modified to output four classes— one for the healthy class label and three for each of the diseases in the dataset.
2. **MobileNet:** A lightweight, mobile CNN architecture called MobileNet was created with the goal of balancing latency and accuracy while outperforming existing state-of-the-art architectures with fewer training parameters. It offers two hyperparameters that enable a user to change the trade-off between latency and accuracy based on the limitations of the situation [40]. It includes 1000 output classes and a 224 × 224 × 3 picture input layer, similar to VGG16. As a result, the MobileNet architecture was altered to suit our needs.
3. **ResNet50:** By reformatting layers as residual learning functions that act with relation to the input layers rather than unreferenced functions, ResNets, also known as Residual Networks, attempt to simplify the difficulty of training deeper neural networks [40] compared to less complicated non- residual designs, residual networks can reach depths that are many times greater. The architects triumphed in the ILSVRC and COCO classification competitions, among others. ResNet50 is a residual network with 50 layers, according to its specifications. Even among the deeper Residual Networks, while being one of the least deep, it performs on par with cutting-edge designs. To assess its performance, a customized the ResNet50 architecture, a 50-layer deep residual network design, was used for the problem at hand.

The last fully connected layer of each model had a SoftMax implementation with 1000 classes since it was pretrained on the ImageNet dataset, which has 1000 classes; we reduced this to 4 classes by altering the fully connected layers suitably. Furthermore, we freeze all levels of the model except from the completely linked layers in order to incorporate the idea of transfer learning as previously mentioned (see section 1 and Fig. 1). This resulted in a large decrease in trainable parameters and, consequently, computational complexity since only the parameters in the fully connected layer were trained and the convolutional layer's parameters were utilized as-it-is for training. Using approved techniques that guarantee data balance, the dataset was randomly divided into training and testing sets for the purpose of proper training and testing. 70% and 30%, respectively, of the

Table 3. Performance comparative statistics with state-of-art models

	VGG16	*MobileNet*	*ResNet50*	*Developed Model*
Layers	16	28	50	6
Parameters	1,51,17,667	64,40,387	3,00,10,499	5,04,71,939
Accuracy	0.9381	0.9434	0.8749	0.9449
Precision	0.9418	0.9456	0.9022	0.9471
Recall	0.9381	0.9487	0.8749	0.9449
F1-Score	0.9372	0.9472	0.8748	0.9445
Average training time	100	30	60	5
Average time taken per image	560	50	206	38

entire dataset were made up of the training and testing datasets, which offered enough pictures for training as well as a sizeable quantity of testing data to produce insightful testing findings. Using the training data and batches of 32 photos at a time, the training was carried out across 10 epochs for each model. With the Adaptive Moment Estimator (ADAM) optimizer and Sparse Categorical Cross-Entropy as the loss function, the pretrained models used were made accessible via the Keras API. The model was then evaluated over the testing dataset to verify its correctness while using fresh data. The comparative performance statistics so obtained are given in Table 3.

MAJOR FINDINGS AND DISCUSSIONS

The major findings of the present research are discussed in this section.

- To categorise the photos as either healthy or having one of the illnesses mentioned as having occurred, each of the models was applied to the compiled dataset. On the basis of the previously mentioned metrics—accuracy, precision, recall, and F1 score—the models were assessed on the test dataset to see how well they performed while trying to classify fresh data. Fig. 3 depicts the performance metric results that have been achieved by the proposed model, having accuracy: '94.49%', precision: '94.71%', Recall: '94.49' and F1-Score: '94.45%', maintaining consistent results across the various performance criteria which were one of the limitations observed in the related works.
- Table 3 shows the comparison between the proposed model and three other models, namely: VGG16, MobileNet and ResNet50. Even though the performance metrics achieved by the proposed model: Accuracy: '94.49%', Precision: '94.71%', Recall: '94.49' and F1-Score: '94.45%', do not significantly differ from those observed in the case of MobileNet : Accuracy: '94.34%', Precision: '94.56%', Recall: '94.87' and F1-Score: '94.72%', under the same circumstances (number of epochs and dataset preparation), the main difference between the two can be observed in the case of average training time – Proposed model: '5 minutes' and MobileNet: '30 minutes', as well as the average time taken per image – Proposed model: '38 milliseconds' and MobileNet: '50 milliseconds', these considerable deviations has been achieved by the overall reduction in the complexity of the proposed model as compared to MobileNet.
- As is evident in Table 3. the proposed model only consists of a minimal '6' layers (nearest being VGG16 with 16 layers) which is considerably lower than any of the other models whilst still maintaining on par or better performance than the other models taken under consideration. This has been attained by tweaking the hyper-parameters, leading to significant reduction in the computational requirements to run the model along with the strides in time for training of the model and classification of the images, thus providing the proposed model with an advantage in the real-time application of the models for day-to-day analysis.

CONCLUSION AND FUTURE SCOPE

This paper proposes a novel model for the early detection of diseases in wheat crops using image analysis. Our model is based on a convolutional neural network that was pre-trained on a plant disease dataset. Notably, our model was shallower than the other models compared, making it more computationally efficient. Through comparative analysis with popular models such as VGG16, MobileNet, and ResNet50, we demonstrated that our proposed model outperformed these models in terms of computational cost, while still achieving comparable or even better results in terms of accuracy, precision, recall, and F1- score, all within the same training time. This suggests that our model is a more efficient and effective solution for this task, especially for devices with limited computational power. Given the impressive performance of our model in terms of computation and time, it has the potential to revolutionize disease diagnosis and treatment in the agricultural industry. By integrating the model with a knowledge base, it can be used to provide real-time remedies for various diseases in their early stages, thereby increasing the annual production of crops such as wheat. Moreover, the computational efficiency of our model makes it suitable for deployment on affordable mobile devices with a camera module, enabling farmers to diagnose crop diseases using their smartphones. This is particularly valuable for farmers in remote areas who may not have access to specialized equipment or experts. While our model has already demonstrated excellent performance, there is still room for improvement. By continuously training the model on an extended dataset, we can further enhance its accuracy and precision. Additionally, we plan to test the model for more diseases and more crops, expanding its potential uses and impact. Overall, our proposed model represents an exciting advancement in the field of disease detection in crops and has the potential to significantly improve crop yields and food security.

REFERENCES

[1] C. Affonso, A.L. Rossi, F.H. Vieira, and A.C. de Carvalho, "Deep Learning for Biological Image Classification," *Expert Systems with Applications*, vol. 85, pp. 114–122, 2017. Doi: 10.1016/j.eswa.2017.05.039

[2] W. Shafik, A. Tufail, A. Namoun, L.C. De Silva and R.A.A.H.M. Apong, "A Systematic Literature Review on Plant Disease Detection: Motivations, Classification Techniques, Datasets, Challenges, and Future Trends," *IEEE Access*, vol. 11, pp. 59174-59203, 2023, doi: 10.1109/ACCESS.2023.3284760

[3] Ramadan, S.T.Y., Sakib, T., Farid, F.A., Islam, M.S., Abdullah, J., Bhuiyan, M.R., Mansor, S., & Karim, H.A. (2024). Improving Wheat Leaf Disease Classification: Evaluating Augmentation Strategies and CNN-Based Models with Limited Dataset. *IEEE Access*, 1. https://doi.org/10.1109/access.2024.3397570

[4] R. Kaundal, A.S. Kapoor, and G.P.S. Raghava, "Machine learning techniques in disease forecasting: A case study on Rice Blast Prediction," *BMC Bioinformatics*, vol. 7, no. 1, 2006, doi: 10.1186/1471-2105-7-485

[5] D. Kim, T.F. Burks, J. Qin, D.M. Bulanon, "Classification of grapefruit peel diseases using color texture feature analysis". *International Journal on Agriculture and Biological Engineering*, vol. 2, no. 3, September 2009, doi: 10.3965/j.issn.1934-6344.2009.03.041-050

[6] B. Kowshik, V. Savitha, M. Nimosh madhav, G. Karpagam, Sangeetha K. "Plant Disease Detection Using Deep Learning", *International Research Journal on Advanced Science Hub*, 3, Special Issue ICARD-2021 3S, 2021, 30-33. doi: 10.47392/irjash.2021.057

[7] G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi, and M. Nithya, "Plant Leaf Disease Classification and Detection System using machine learning," *Journal of Physics: Conference Series*, vol. 1712, no. 1, p. 012012, 2020, doi: 10.1088/1742- 6596/1712/1/012012

[8] A.H. Kulkarni and A. Patil (2012), "Applying image processing technique to detect plant diseases", vol 2. 3661-3664.

[9] A. Hussain, M. Ahmad, I.A. Mughal, and H. Ali, "Automatic Disease Detection in Wheat Crop using Convolution Neural Network," *The 4th International Conference on Next Generation Computing 2018*, Dec. 2018, doi: 10.13140/RG.2.2.14191.46244

[10] B. Liu, Y. Zhang, D.J. He, and Y. Li, "Identification of Apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, 2017, doi: 10.3390/sym10010011

[11] A. Khan, A. Sohail, U. Zahoora, and A.S. Qureshi, "A survey of the recent architectures of deep convolutional Neural Networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6

[12] A. Patel and M.B. Joshi, "A survey on the plant leaf disease detection techniques", *IJARCCE*, pp. 229– 231, 2017, doi: 10.17148/ijarcce.2017.6143

[13] S. Ghosal, D. Blystone, A.K. Singh, B. Ganapathy-subramanian, A. Singh, and S. Sarkar, "An explainable deep machine vision framework for Plant Stress Phenotyping," *Proceedings of the National Academy of Sciences*, vol. 115, no. 18, pp. 4613–4618, 2018, doi: 10.1073/pnas.1716999115

[14] K. Nagasubramanian, S. Jones, A.K. Singh, S. Sarkar, A. Singh, and B. Ganapathysubramanian, "Plant disease identification using explainable 3D deep learning on hyperspectral images," *Plant Methods*, vol. 15, no. 1, 2019, doi: 10.1186/s13007-019-0479-8

[15] J. Kaur, R. Chadha, S. Thakur, R. Kaur, "A Review Paper on Plant Disease Detection using Image Processing and Neural Network Approach". *International Journal of Engineering Sciences & Research Technology*. April 2016. ISSN: 2277-9655.

[16] M. Kumar, P. Gupta, P. Madhav and Sachin, "Disease Detection in Coffee Plants Using Convolutional Neural Network", *2020 5th International Conference on Communication and Electronics Systems* (*ICCES*), Coimbatore, India, 2020, pp. 755-760, doi: 10.1109/ICCES48766.2020.9138000.

[17] A.M. Reshmi, P. Prasidhan, "Leaf Disease Detection Using CNN", *International Journal of Creative Research Thoughts* (*IJCRT*), Volume 10, Issue 3, pp. 235-f241, March 2022, ISSN:2320-2882

[18] Md. Tariqul Islam, "Plant Disease Detection using CNN Model and Image Processing", *International Journal of Engineering Research & Technology* (*IJERT*), Volume 09, Issue 10, October 2020, doi: 10.17577/IJERTV9IS100123

[19] W. Zeng, M. Li, J. Zhang, L. Chen, S. Fang, and J. Wang, "High-order residual convolutional neural network for robust crop disease recognition," *Proceedings of the 2nd International Conference on*

Computer Science and Application Engineering, 2018, doi: 10.1145/3207677.3277952

[20] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense," *Journal of Sensors*, vol. 2019, pp. 1–13, 2019, doi: 10.1155/2019/7630926

[21] X. Zhang, L. Han, Y. Dong, Y. Shi, W. Huang, L. Han, P. González-Moreno, H. Ma, H. Ye, and T. Sobeih, "A deep learning-based approach for automated yellow rust disease detection from high-resolution hyperspectral UAV images," *Remote Sensing*, vol. 11, no. 13, p. 1554, 2019, doi: 10.3390/rs11131554

[22] D. Wang, R. Vinson, M. Holmes, G. Seibel, A. Bechar, S. Nof, and Y. Tao, "Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (or-ac-gan)," *Scientific Reports*, vol. 9, no. 1, 2019, doi: 10.1038/s41598-019-40066-y

[23] Sahu, Madhusmita & Dash, Rasmita (2021). A Survey on Deep Learning: Convolution Neural Network (CNN). 10.1007/978-981-15-6202-0_32.

[24] S. Albawi, T.A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network", 2017, *International Conference on Engineering and Technology* (*ICET*), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[25] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* 9, 611– 629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[26] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks: A Review," arXiv.org, 16-Sep-2020. [Online] Available: https://doi.org/10.48550/arXiv.2009.07485

[27] G. Lin and W. Shen, "Research on convolutional neural network based on improved ReLU piecewise activation function," *Procedia Computer Science*, vol. 131, pp. 977–984, 2018, doi: 10.1016/j.procs.2018.04.239CNN and Softmax [Online] Available: https://www.andreaperlato.com/aipost/cnn-and-softmax/

[28] L. Goyal, C. Sharma, A. Singh, and P. Singh, "Leaf and spike wheat disease detection & classification using an improved deep convolutional architecture," *Informatics in Medicine Unlocked*, vol. 25, p. 100642, 2021, doi: 10.1016/j.imu.2021.100642

[29] W. Chen, C. Wellings, X. Chen, Z. Kang, and T. Liu, "Wheat stripe (yellow) rust caused bypuccinia striiformisf. sp.tritici," *Molecular Plant Pathology*, vol. 15, no. 5, pp. 433–446, 2014, 10.1111/mpp.12116.

[30] Y. Anikster, J. Manisterski, D.L. Long, and K.J. Leonard, "Leaf Rust and stem rust resistance in triticum dicoccoides populations in Israel," *Plant Disease*, vol. 89, no. 1, pp. 55–62, 2005, doi: 10.1094/pd-89-0055

[31] M. Bolton, J. Kolmer, and D. Garvin, "Wheat leaf rust caused bypuccinia triticina," *Molecular Plant Pathology*, vol. 9, no. 5, pp. 563–575, 2008, doi: 10.1111/j.1364-3703.2008.00487.x

[32] J. Huerta-Espino, R.P. Singh, S. Germán, B.D. McCallum, R.F. Park, W.Q. Chen, S.C. Bhardwaj, and H. Goyeau, "Global status of wheat leaf rust caused by puccinia triticina," *Euphytica*, vol. 179, no. 1, pp. 143–160, 2011, doi: 10.1007/s10681-011- 0361-x

[33] W. Spielmeyer, R.A. McIntosh, J. Kolmer, and E.S. Lagudah, "Powdery mildew resistance and LR34/YR18 genes for durable resistance to leaf and stripe rust cosegregate at a locus on the short arm of chromosome 7d of wheat," *Theoretical and Applied Genetics*, vol. 111, no. 4, pp. 731–735, 2005, doi:10.1007/s00122-005-2058-9.

[34] S. E. Strelkov and L. Lamari, "Host–parasite interactions in tan spot [pyrenophora tritici-repentis] of wheat," *Canadian Journal of Plant Pathology*, vol. 25, no. 4, pp. 339–349, 2003, doi: 10.1080/07060660309507089.

[35] A. Shabeer, "Tan spot effects on yield and yield components relative to growth stage in winter wheat," *Plant Disease*, vol. 72, no. 7, p. 599, 1988, doi: 10.1094/pd-72-0599

[36] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," *Lecture Notes in Computer Science*, pp. 345–359, 2005, doi: 10.1007/978-3-540-31865-1_25.

[37] S. Sun, Z. Cao, H. Zhu and J. Zhao, "A Survey of Optimization Methods from a Machine Learning Perspective" in *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3668-3681, Aug. 2020, doi: 10.1109/TCYB.2019.2950779.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv.org, 10-Apr-2015. [Online]. Available: https://arxiv.org/abs/1409.1556.

[39] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for Mobile Vision Applications," arXiv.org, 17-Apr-2017. Available: https://arxiv.org/abs/1704.04861

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016, doi: 10.1109/cvpr.2016.90.