Wheat Rust Disease Detection and Classification using an improved Deep Learning Algorithm

Bader Ijaz

b.ijaz@studenti.uniss.it

University of Sassari: Universita degli Studi di Sassari https://orcid.org/0000-0002-6298-0644

Khalid Hamza

Saba Un Nisa

Muhammad Saeed

Badar Ijaz

Research Article

Keywords: Agricultural automation, Deep learning, Image classification, Real time dataset, Wheat crop, Wheat diseases

Posted Date: January 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8460561/v1

Wheat Rust Disease Detection and Classification using an improved Deep Learning Algorithm

Khalid Hamza[1] · Saba Un Nisa[2] · Muhammad Saeed[3] · Badar Ijaz[4]

[1]Department of Plant Pathology, PMAS Arid Agriculture University, Rawalpindi 46300, Pakistan

[2]University Institute of Information Technology, PMAS Arid Agriculture University, Rawalpindi 46300, Pakistan

[3]Wheat Research Station, Murree 47150, Pakistan

[4]Department of Agricultural Sciences, University of Sassari, Viale Italia 39/A, 07100 Sassari, Italy.

Corresponding author: Badar Ijaz

Email address: b.ijaz@studenti.uniss.it, baderzaildar@gmail.com (B. Ijaz)

ORCID:0000-0002-6298-0644

Abstract

Wheat, the third most widely consumed cereal crop worldwide, faces substantial yield and quality losses as a result of rust disease, notably leaf rust, stem rust, and stripe rust. These rust disease, caused by *Puccinia triticina*, *Puccinia graminis*, and *Puccinia striiformis*, respectively, are capable of causing significant yield losses in wheat in the absence of timely detection. Conventional disease identification relies heavily on manual visual inspection, which is time consuming, labor intensive, and prone to error, especially in large scale agricultural systems. To address these limitations, this study proposes a deep learning-based framework for the early detection and classification of wheat rust diseases. A real-time dataset was developed using field images collected from various wheat-growing regions and augmented with publicly available data. The dataset comprises images of healthy leaves and those affected with the three major rust diseases. A modified convolutional neural network (CNN) architecture was employed for extract features and disease classification. Experimental results demonstrate that the proposed approach achieves high classification accuracy, highlighting its effectiveness as a reliable tool for automated wheat rust detection in precision agriculture. By enabling rapid and accurate disease identification, the system supports timely decision-making, reduces potential yield losses, and improves crop management practices, thereby contributing to food security and sustainable agricultural production.

Keywords Agricultural automation · Deep learning · Image classification · Real time dataset · Wheat crop · Wheat diseases

Introduction

Wheat is one of the most important cereal crops in the world, ranking third in consumption after maize and rice. Despite its importance for food security, wheat production remains vulnerable to biotic stresses, of which fungal diseases are the most significant. According to the Food and Agriculture Organization of the United Nations, approximately one-tenth of the world's population suffers from food insecurity and malnutrition, while crops in developing nations produce disproportionately lower harvests compared with those grown in more developed regions (Onyeaka et al., 2024). The inferior yields of such crops have often been attributed to a general lack of adoption of modern agricultural practices and underdeveloped systems of disease surveillance.

Wheat is the main staple crop and one of the most essential contributors to national food security, rural livelihoods, and economic stability in Pakistan (Rasheed et al., 2024). About 80% of smallholder farmers depend on the wheat sector, which is a significant contributor to GDP. (Shahzad et al., 2022); (Chandio et al., 2023) Wheat productivity in Pakistan is often hampered by diseases of this plant species, impacting not only the quantity but also the quality of grains. Disease monitoring with timely intervention thus assumes prime importance for sustainable wheat production and rural development (Hussain & Maharjan, 2025).

Among wheat diseases, rusts leaf rust (*Puccinia triticina*) (Krępski et al., 2024), stripe rust (*Puccinia striiformis*) (Shahin et al., 2024), and stem rust (*Puccinia graminis*) (Lubega et al., 2024) are considered the most destructive worldwide. These obligate fungal pathogens exhibit complex life cycles and can rapidly spread under favorable environmental conditions, infecting leaves, stems, and spikes (Figueroa et al., 2018); (Chai et al., 2020). Leaf rust is characterized by small reddish-brown pustules on leaf surfaces, stripe rust forms yellow linear pustules along leaf veins, while stem rust produces elongated lesions on stems and spikes (Krępski et al., 2024); (Shahin et al., 2024); (Lubega et al., 2024). Rust infections severely disrupt photosynthesis and grain filling, resulting in substantial yield losses when not detected at an early stage (Ray, 2024).

Traditional wheat rust diagnosis heavily depends on visual field inspections by trained personnel (Chandio et al., 2023). Though this is effective at the small scale, the approach is time-consuming, subjective, and not feasible for monitoring large-scale areas. Delays or incorrect disease identification may result in inappropriate control measures, higher production costs, and significant yield loss. (Ahmad et al., 2023) and (Maulenbay & Rsaliyev, 2024) point out that there is an emerging need for automated, quick, and realiable disease detection systems which may help in timely plant protection decisions.

Recent advances in digital agriculture and image-based analysis have enabled the application of machine learning and deep learning techniques for plant disease detection (Yap & Al-Mutairi, 2024). Among them, CNNs have been particularly popular for their excellent performance in recognizing disease-specific visual symptoms in crop images (Wani et al., 2022) (Ahmad et al., 2023). Various research works have successfully implemented CNN-based models on detecting wheat rust (Verma et al., 2024) from both field, UAV, and smartphone images (Kumar et al., 2024). 3D-CNN approaches, transfer learning through VGG, ResNet, and EfficientNet architectures, and several ensemble learning strategies have reported high classification accuracies (Jagadeeshan et al., 2024); (Nigam et al., 2023); (Tola et al., 2024); (Reis & Turk, 2024). This has further been enhanced by data augmentation methodologies with generative adversarial networks under limited dataset conditions (Ramadan et al., 2024).

Despite these encouraging results, several limitations in the existing studies from a plant protection viewpoint exist. Most models are trained on region-specific or secondary datasets, which limit their applicability across diverse agro-ecological conditions. Others use a complex hybrid pipeline or computationally intensive architecture and therefore may not see practical deployments in real farming environments. In addition, real-time field conditions and reproducibility of the works are not emphasized enough to ensure their adoption in regular crop protection practices.

The current study introduces an image-based deep learning framework for early detection and classification of major wheat rust diseases in realistic field conditions. The system is developed on a newly constructed dataset of wheat leaf images, which was collected from multiple wheat-growing regions and supplemented with publicly available data. In this respect, the study proposes a revised convolutional neural network architecture for precise discrimination among leaf rust, stripe rust, stem rust, and healthy plants. The proposed approach will enable early and accurate identification of diseases to aid timely plant protection measures for reducing yield losses in environmentally friendly wheat production systems.

Materials and Methods

To classify images and detect plant diseases, a variety of deep learning architectures may be used (Fang et al., 2023). We briefly describe the VGG16 and ResNet 50 architectures in this section, two of these well-liked architectures. These designs have mostly been used to address issues with visual object recognition. Using transfer learning and parameter changes, both networks can be utilized for tasks such as video classification, image indexing, region of interest extraction, semantic segmentation and retrieval. ResNet 50 and VGG-16 were used to compare when classifying the experimental outcomes for wheat disease.

Deep learning architecture VGG-16

VGG16 uses 16 weighted layers of convolutional neural networks (CNN) to construct its deep learning architecture. Convolutional layers, max-pooling layers, activation layers and similar types are included in the classifications for these layers. In addition, the network includes 13 convolutional layers, 5 layers with maximum pooling and 3 connected layers. There are only 16 weight layers in the architecture. A SoftMax classifier is formed by two 4096-node layers that are fully connected to each other. After each pooling layer, the network's breadth increases by a factor of 2, starting with 64. VGG16 (Reis & Turk, 2024) however, the model has 140 million parameters. The VGG16 model which was trained on the ImageNet dataset, can be used to approach similar image identification tasks.

Deep learning architecture RESNET-50

ResNet 50 is a different idea for designing neural networks with 50 additional layers (Tola et al., 2024). ResNet's main goal is to demonstrate a 'characteristic alternative connection method' that doesn't require any additional layers. There are 23,587,712 parameters in all in the ResNet50 standard model. There are 53,120 non-trainable parameters and 23,534,592 trainable ones.

Proposed model

In this study we used automated methods for the early detection of wheat diseases. We use convolutional neural networks (CNNs) that detect wheat disease; data accessibility becomes an essential part of CNN training. Many researchers have used CNNs to diagnose wheat diseases because of their accuracy in categorizing and recognizing images. Our proposed approach's complete workflow is illustrated in (Fig. 1), which includes evaluation using our own self-generated database. Training and validation tests were being carried out with pre-trained CNNs and enhanced VGG model. Through convolutional and pooling layers, these pre-trained CNNs can extract valuable features for image classification by leveraging the spatial correlations among the pixels in the image.

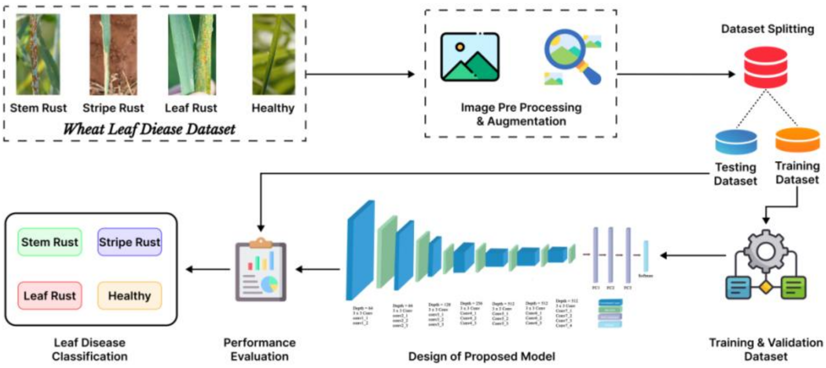

Fig.1 Classification methodology for wheat rust disease

Dataset description

Field data were collected in Pakistan from the North Punjab (Fig. 2) in the month of February 2025 to March 2025. The majority of the data was collected in the field as defined in (Table 1), while the remaining data was collected via publicly available images that were located in the Kaggle images database (*Wheat Plant Diseases*, n.d.). Sampling was done in important wheat-growing locations to observe how common and serious rust diseases are. We note down field views, assign an intensity to the symptoms, mark the area with GPS, write the wheat variety and add temperature and humidity data. We collected these samples at the tillering to heading stages of wheat growth. The reason for this dataset is to assist study of disease distribution, find ways to breed rust-free varieties and make early detection possible. Deep learning models can also be developed with these data to automatically find diseases. The information in this data helps plant pathologists, agronomists and policymakers respond to problems with wheat rust in the area.

In this study, a unified dataset was constructed by combining real-time field images and relevant images retrieved from Google Images. This approach aimed to increase the diversity and size of the dataset while maintaining relevance to real-world disease symptoms. All images were preprocessed uniformly to eliminate bias from different sources. Although we do not perform separate performance analysis for field and Google datasets, the integrated dataset was used to train and evaluate the proposed model. This ensured better generalization across varying lighting, backgrounds, and image quality.

To compare the quality and consistency of both sources, we analyzed image resolution, lighting conditions, background clutter, and label confidence. Field images had more variability in environmental conditions but were captured with precise metadata e.g., GPS, variety, humidity. In contrast, online images were generally cleaner but lacked contextual information. Both sets were pre-processed uniformly to ensure consistency before training. Classification performance was evaluated using combined data to maximize model generalization.

Table 1 Dataset description

Attribute	Description
Rust Type	Leaf Rust, Stem Rust, Stripe Rust
Collection Region	North Punjab, Pakistan
Collection Period	Feb–March 2025
Collection Method	Field observation + Kaggle image samples
Plant Growth Stage	Tillering to heading
Field Data Includes	Disease severity, GPS, wheat variety, weather data
Usage	Disease classification, model training, analysis
Repository Link	https://www.kaggle.com/datasets/sabaunnisa/wheat-rust-disease

We have divided the datasets 1294 into 962 training images and 332 testing images. In the current study, a 3:1 ratio was used to create the training, and validation sets for the image dataset, meaning 75% of the data 722 used to training and 25% 240 to validation. A fixed random seed (`seed = 42`) was used to ensure reproducibility and stratified sampling preserved class balance across splits. Additionally, to reduce overfitting and improve generalization. We applied data augmentation techniques. The distribution of the dataset by class is illustrated in (Fig. 3).

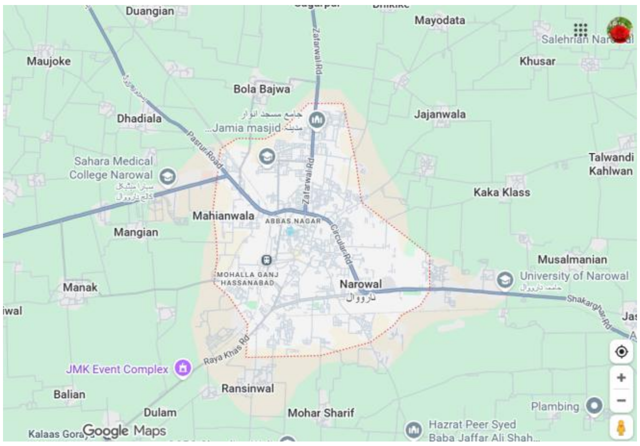

Fig. 2 North Punjab Region

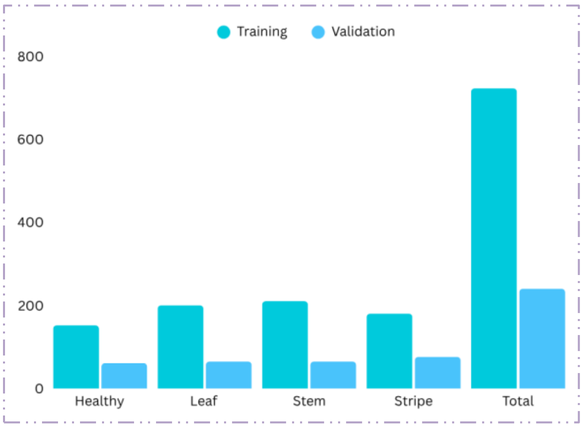

Fig. 3 Class wise distribution of rust dataset

Pre-processing of data

To ensure model robustness and consistency, a comprehensive image pre-processing pipeline was applied before training. The primary goal of pre-processing was to normalize the dataset across variations in lighting, scale, orientation, and noise. All images were resized to 224 × 224 pixels to match the input requirements of standard CNN architectures such as VGG. Irrelevant backgrounds were cropped to focus on the diseased regions of the leaves. Image sharpening techniques were applied to enhance edge definition and leaf texture, while contrast enhancement improved the visibility of disease symptoms. Brightness adjustments were made to normalize lighting conditions, and mild Gaussian blur was introduced to reduce noise while preserving essential image features. Finally, all images were converted to RGB color format and saved in JPEG (.jpg) format to ensure uniformity across the dataset. These pre-processing steps significantly improved image quality and consistency, thereby contributing to higher model training efficiency and accuracy.

Data augmentation

More data was generated through data augmentation to make sure over fitting did not happen. These techniques strengthened how well the model works in new situations and covered more types of data. Among the applied techniques were horizontal flipping, rotating images by 30 degrees, a height shift of 0.2, a width shift of 0.2, zooming and shearing by 0.15. (Fig. 4) provides a sample of each class.

Fig. 4 Class wise dataset

Methodologies used to trained proposed model

The architecture of the proposed model, which was used in the current experiment, is shown in (Fig. 5). This model has 24 layers comprised of 21 convolutional layers and 3 fully connected layers. Addition layers are block 6 and block7 and the equations are (1), (2), (3), (4), (5), (6), (7), (8) and (9). The images included in this model are processed at 224 × 224, which is normal size. To extract feature maps that are uniform or equal, it is necessary to adjust the fixed image size after the convolution process. This model employed a kernel with a fixed size of 3 × 3. The kernel is sometimes referred to as a filter that is in charge of removing features from the input images. Vertical edges, horizontal edges, or a combination of both can be included in these retrieved patterns or features. The features have first been extracted using a convolutional technique, and then the classification has been completed. In order to extract features, the kernel/filter in the convolution process traverses the image from the top left to the bottom right corner.

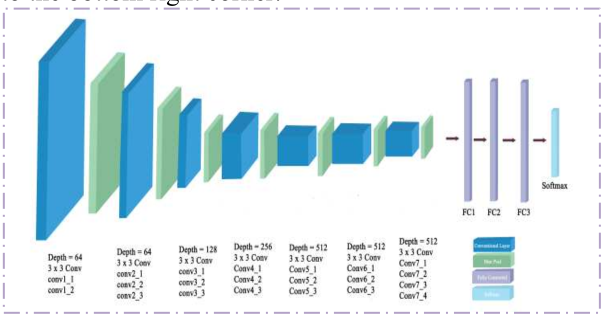

Fig. 5 The architecture of proposed model for wheat rust disease detection.

The kernel's movement is moved by two pixel values when the stride value is 2. In order to further identify things, *Conv* (Convolution) layers are utilized to extract patterns or features from images. Edges and other generic features are extracted in the first layer, and domain-specific features are extracted in the following levels. Each convolution block is followed by the *Maxpool (M*ax-pool layer), which is used for down sampling the feature maps. By keeping the most apparent feature, the dimensionality of the image is decreased in this procedure. Several feature maps are generated as a result of the convolution layers. These feature maps are further flattened and mapped with a fully connected layer in the categorization module. The model in this case has a dense layer and a feature vector. The following dense layer of the same size receives this feature vector in addition. Utilizing the soft-max activation function, the last layer neurons are finally fully coupled to the output neurons. However, we concentrated on the classification problem with the four classes in our study. The output layer was changed to four classes as a consequence, utilizing the soft-max probability function. Data is transmitted forth and backward before actual learning occurs. The input neurons are multiplied by the weight values in the forward pass, and the $ReLU$ (Rectified Linear Unit) activation function is also employed. After passing through the ReLU activation function, all of the negative pixel values turn positive, which gives the model some additional nonlinearity. In contrast, the backward pass employs back-propagation to lower the loss value. In this process, biases and weights are updated from the final to the first layer, and gradients at each layer are produced using a convolution operator. There are a total of 25,305, 356 parameters in this model. This is significant since transfer learning was employed to build the model using previously trained weights, leaving the remaining parameters un-trainable. Consequently, the model is trained more quickly and with fewer parameters.

$$X_1 = ReLU(Conv_{3X3}^{512}(X)) \quad (1)$$

$$X_2 = ReLU(\ Conv_{3X3}^{512}(X_1)) \quad (2)$$

$$X_3 = ReLU(Conv_{3X3}^{512}(X_2)) \quad (3)$$

$$X_4 = Maxpool_{2X2}(X_3) \quad (4)$$

$$X_5 = ReLU(\ Conv_{3X3}^{512}(X_4)) \quad (5)$$

$$X_6 = ReLU(Conv_{3X3}^{512}(X_5)) \quad (6)$$

$$X_7 = ReLU(Conv_{3X3}^{512}(X_6)) \quad (7)$$

$$X_8 = ReLU(Conv_{3X3}^{512}(X_7)) \quad (8)$$

$$X_9 = Maxpool_{2X2}(X_8) \quad (9)$$

$Conv_{3X3}^{512}$ denotes a 2D convolution layer with a 3×3 kernel and 512 filters.

$$ReLU(x) = max(0, x) \quad (10)$$

is the Rectified Linear Unit activation function.

$MaxPool\ 2 \times 2$ denotes a max pooling operation with a 2×2 window.

The ReLU activation function (10) is applied in each convolutional layer to introduce non-linearity by outputting the input directly if positive and zero otherwise. It enhances learning efficiency and mitigates the vanishing gradient problem. The *Softmax* (Soft-max activation function) (11) is used in the final classification layer to convert raw scores into a probability distribution across target classes, ensuring the outputs sum to one for effective multi-class classification (Majumdar et al., 2015).

$$Softmax(zi) = \frac{e^{zi}}{\sum_{j=1}^{K} e^{zj}}\ for\ i\ =\ 1,2,...,K \quad (11)$$

The decision to use 21 convolutional layers was made after empirical testing with various architectures. Simpler CNNs with fewer layers e.g., 6, 9, or 13 convolutional layers were also implemented and tested on the same dataset, but they failed to capture the complex texture patterns of wheat rust symptoms as effectively, resulting in lower accuracy and F1-scores.

The deeper model allows hierarchical extraction of both low-level and high-level features across layers, improving performance on tasks with fine-grained visual differences such as between stripe rust and leaf rust. Each convolutional block captures increasingly abstract information which, combined with pooling and dense layers, enables more accurate decision boundaries. The added layers Block 6 and Block 7 were specifically designed to deepen the network without drastically increasing computational cost due to parameter freezing transfer learning. Hence, our model achieves high accuracy while maintaining generalization capabilities.

To ensure robust model performance and fair evaluation, we employed a hold-out validation strategy. The dataset was initially divided into 962 training images and 332 testing images. From the training set, 25% is 240 images were used as a validation set for hyper parameter tuning e.g., learning rate, batch size, number of epochs. The split was stratified to maintain class distribution across all subsets. A fixed random seed (`seed = 42`) was applied to ensure reproducibility. Hyper parameters were tuned based on validation performance using early stopping and learning rate scheduling. Accuracy, precision, recall, and F1-score were monitored on both validation and test sets. We report the average results from three independent training runs to ensure statistical robustness.

While the proposed architecture is inspired by VGG-style deep CNNs, we also considered modern, lightweight alternatives such as MobileNetV3 and *SE* Squeeze and Excitation blocks**.** However, preliminary experiments with these models yielded slightly lower accuracy and less robust feature extraction for the wheat rust dataset. SE blocks were also tested as attention modules to improve channel interdependencies, but the added complexity did not significantly improve performance in our case. As wheat rust symptoms often involve subtle texture changes, deeper convolutional layers as in our design were more effective at capturing hierarchical features. Nevertheless, we acknowledge that these alternatives may offer advantages in mobile or embedded deployment scenarios, and we suggest this as a potential direction for future work.

The proposed algorithm 1 is an enhanced VGG-based convolutional neural network designed for the classification of wheat rust diseases.

Algorithm 1 Enhanced VGG16-Based Convolutional Neural Network for Rust Disease Classification

1. **Input:** Image dataset D with categories: healthy, leaf rust, stem rust, stripe rust
2. **Output:** Trained VGG-based classification model
3. **Step 1**: Data Pre-processing
4. **for** each image I in dataset D **do**
5. Resize I to 224 x 224
6. Normalize I using mean pixel values: [123.68, 116.779, 103.939]
7. Append processed image to data array
8. Extract label from directory structure and append to labels
9. **end for**
10. Encode labels using one-hot encoding

11. Split dataset into training and testing sets with 75% and 25% ratio
12. **Step 2: Data Augmentation**
13. Apply transformations: rotation, zoom, width/height shift, shear, flip
14. **Step 3: Model Construction**
15. Load pre-trained VGG16 model without top layers
16. Freeze VGG16 base layers
17. Add custom convolutional blocks:
    - Block 6: 3 Conv layers (512 filters) + MaxPooling
    - Block 7: 3 Conv layers (512 filters) + MaxPooling
18. Add classification head:
    - Global Average Pooling
    - Flatten
    - Dense (512 units, ReLU)
    - Dropout (0.4)
    - Dense (softmax output for 4 classes)
19. **Step 4: Model Compilation and Training**
20. Compile model using Adam optimizer and categorical cross-entropy loss
21. Train model for 80 epochs using augmented data
22. **Step 5: Model Evaluation**
23. Predict labels on test set
24. Generate classification report and confusion matrix
25. Plot training vs validation accuracy and loss
26. Compute and plot multi-class ROC curve
27. **Step 6: Save Outputs**
28. Save the trained model and label encoder to disk

It starts with the pre-processing of the image dataset, which includes resizing the images to 224×224, normalization using standard mean pixel values, and label extraction. The dataset is then divided into two subsets: training and testing. Training data is further augmented by applying random transformations such as rotation, zoom, shift, and flip in order to enhance model generalization. The base consists of a pre-trained VGG16 model whose convolutional layers are frozen, followed by the addition of extra convolutional blocks (Block 6 and Block 7) to deepen the architecture. A custom classification head consisting of Global Average Pooling, dense, and dropout is added at the end, terminated with a soft-max layer to perform multi-class classification. The model is compiled with the Adam optimizer and categorical cross-entropy loss, then trained for 80 epochs. Finally, after training is performed, the model will be evaluated using metrics such as classification, confusion matrix, and ROC curves. The model trained and label encoder are then saved for deployment.

Hyper-tuning of parameters

Hyper-parameters like learning rate, batch size, loss function, and number of epochs are typically taken into account when fine-tuning the model. Each hyper-parameter is taken into consideration within a certain range to develop the classification model for four classes. In order to create an effective model, numerous experiments have been carried out. Following some testing with different values for the specified hyper-parameters, significantly affect the model's accuracy. The following hyper-parameters have been used in the current study, as stated in (Table 2). The model obtained high classification accuracy with these parameters.

Hyper parameter values were selected through manual testing and iterative evaluation of model performance. In future work, automated search techniques such as grid search or random search could be explored to further optimize the model's performance.

Table 2 Parameters used in the training process

Parameter Name	Parameter Value
Batch size	10
Loss Function	categorical cross-entropy
Optimizer	Adam
Epochs	80
Initial learning rate	$1\ e^{-3}$

Experiment setup

The implementation and training of both the existing and proposed models were carried out using Google Colab Notebooks, which provide access to GPU-accelerated environments for deep learning tasks. Specifically, we utilized NVIDIA Tesla T4 GPUs, which are commonly offered in Colab's free tier. These GPUs significantly reduce training time and support high-performance computations required for convolutional neural networks.

The models were trained for 80 epochs**,** and performance was evaluated on their ability to classify images across multiple disease categories. GPU acceleration enabled efficient handling of large datasets and deep architectures, improving both training speed and resource utilization. The generalization ability of the models was validated using standard performance metrics such as accuracy, precision, recall, and F1-score on the test set.

Results and discussion

Accuracy and loss results

The model has been trained using an image dataset for the classification of wheat diseases. We utilized the online, GPU-compatible Google Colab platform. We examine the best experiment, which yields the maximum training accuracy, out of the ones that were conducted. The training and validation accuracy results at various epochs (ranging from 10 to 80) are shown in (Fig. 6) along with the associated loss values. The training accuracy in this case starts at 86.87% after 10 epochs and rises to 98.23% after 80 epochs. We kept on computing the accuracy for the further 90 epochs, but we saw no appreciable gain in training accuracy. Even though more experiments might be run by upping the epoch count, the accuracy at epoch 80 was quite encouraging. On the other hand, (Fig. 7) demonstrates that the training and validation accuracy and loss varied between proposed model, VGG16 and Resnet50. Similar to this, it was found that the training loss diminishes at every higher epoch step (between 10 and 80).

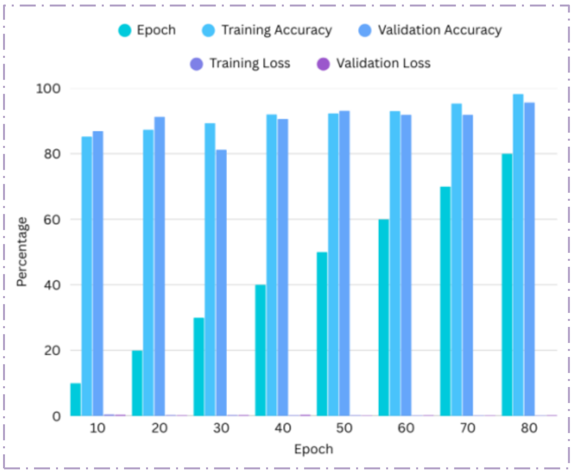

Fig. 6 Training and validation accuracy and loss

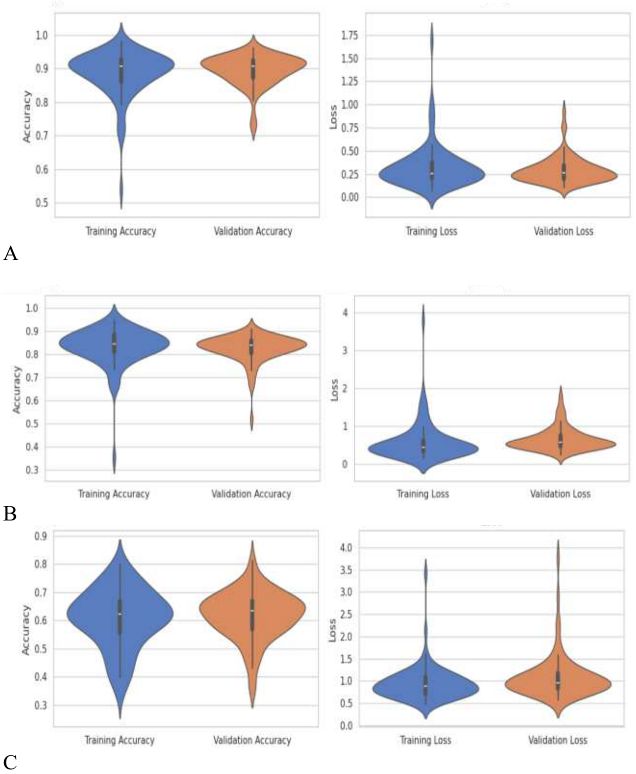

Fig. 7 The accuracy and loss A,B, C Proposed Model, VGG16 and Resnet 50

Model evaluation

We conducted test trials on the validation data (which is 25% of the total dataset) to assess how well the trained model performed. In this way, a total of 76 representative images of stripe rust, 97 representative images of stem rust, 69 representative images of healthy leaves, and 90 images representing leaf rust have been taken into consideration. A confusion matrix, (Fig. 8), was used to evaluate the testing results. There are various metrics used to measure performance, including accuracy, precision (P), recall (R), F1-score, and accuracy (Li et al., 2023). Each of the performance metrics is checked using prediction labels. Using TP, TN and FP, each performance metric can be calculated and every term is described in (Table 3). Equations for precision, recall and F1-score are shown in (12), (13) and (14). (Fig. 9) illustrates the precision, recall, F1-score, accuracy and how much other intra-classes are similar to this class.

The multi-class classification model's performance is evaluated using *ROC* (Receiver Operating Characteristic) curve. The *TPR* (True Positive Rate) and *FPR* (False Positive Rate) are plotted against various threshold settings for each class. The ability of the model to distinguish between classes is indicated by the *AUC* (Area Under the Curve); a higher AUC indicates better classification performance. (Fig. 10) shows that in this study, ROC curves are generated for all wheat rust classes to visualize and compare the predictive strength of the enhanced VGG16 model.

$$P = \frac{Tp}{(Tp+Fp)} \tag{12}$$

$$R = \frac{Tp}{(Tp+Fn)} \tag{13}$$

$$F1 = \frac{2*P*R}{(P+R)} \tag{14}$$

Misclassification analysis

To better understand the limitations of the proposed model, we conducted a misclassification analysis by examining instances where the model's predictions did not match the ground truth. Notably, most misclassifications occurred between leaf rust and stripe rust, which share similar visual symptoms in early stages such as yellowish streaks or irregular lesion shapes. In low-contrast or shadowed images, these features become ambiguous, leading the model to confuse one for the other. For example, Case 1 is a leaf infected with early-stage stripe rust was incorrectly predicted as leaf rust. The visual symptoms were minimal and lacked the distinct linear alignment characteristic of mature stripe rust. Case 2 is a highly magnified image of leaf rust with background noise was misclassified as healthy due to sparse lesion patterns resembling natural leaf texture. These failure cases suggest a need for: Enhanced data augmentation**,** especially for lighting and occlusion conditions. Region-based attention mechanisms to help the model focus on biologically relevant lesion zones. Fine-grained annotation or bounding boxes to improve feature localization.

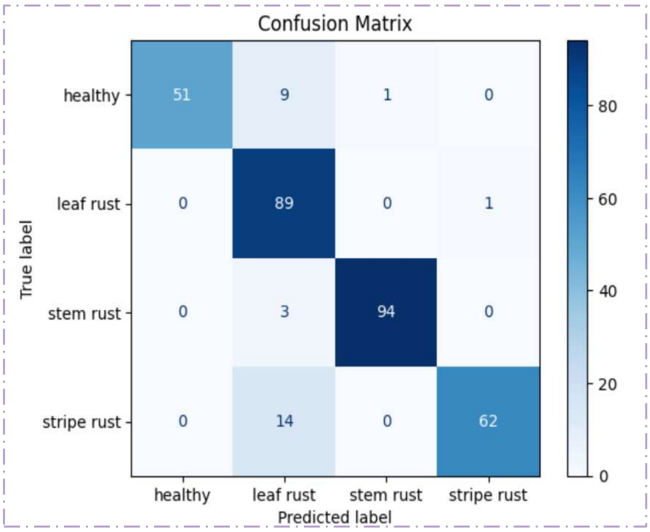

Fig. 8 The proposed model confusion matrix

Table 3 Description of performance parameter

Parameter	Description
Tp	Wheat rust pixels are found in the patches that were detected. (Liu, 2020)
Fp	Patches that do not show true wheat rust in their bounding boxes.(Liu, 2020)
Fn	Number of wheat real rust pixels that are not included in the identified patches. (Liu, 2020)

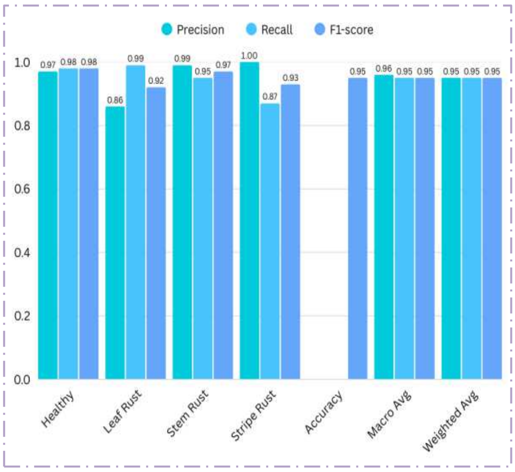

Fig. 9 Measures of the proposed method's accuracy

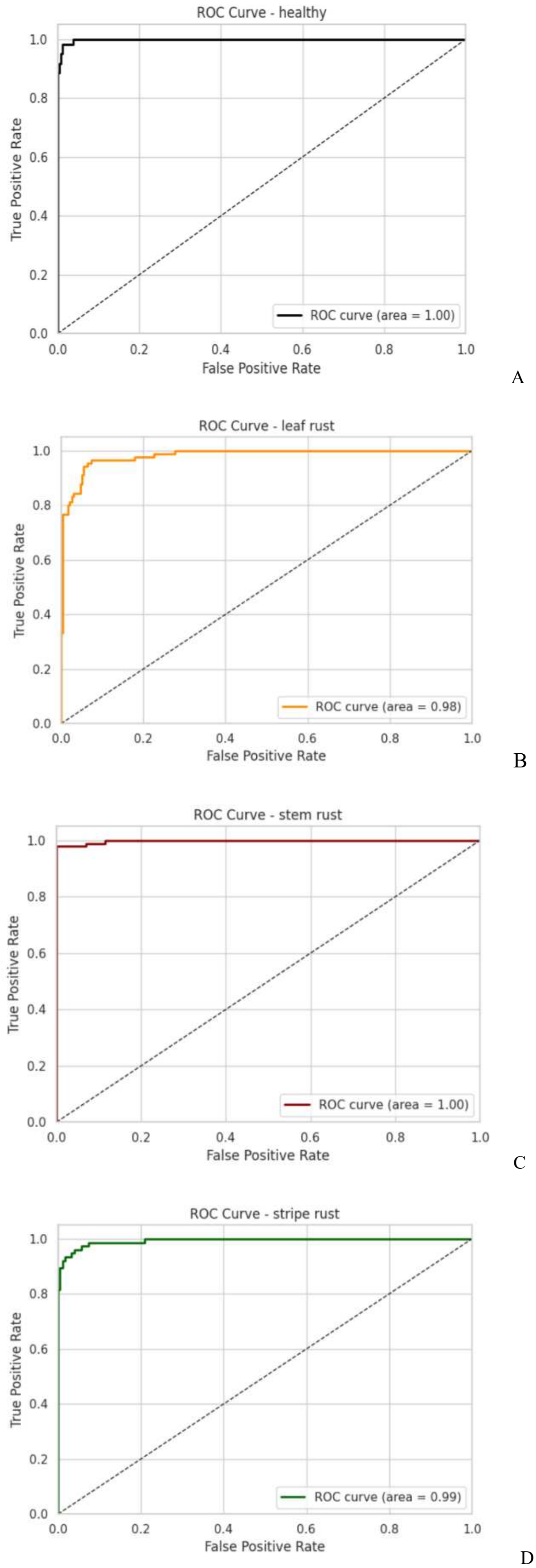

Fig. 10 ROC Curve A,B,C,D Healthy, leaf rust, stem rust, stripe rust

Ablation study

To evaluate the contribution of various design choices, we conducted ablation experiments across four key areas A detailed ablation study analyzing the impact of added blocks, batch size, regularization, and augmentation is provided in (Table 4).

Table 4 Ablation study

Experiment	Configuration	Accuracy (%)	Observation
Baseline Model	Without Block 6 & 7	92.3	Accuracy drops, less deep feature extraction
+ Block 6 & 7	Final model with added blocks	95.0	Significant boost due to deeper representations
Batch Size = 8	Smaller mini-batch	93.1	Slightly unstable convergence
Batch Size = 10	Final selected batch size	95.0	Best trade-off of stability & accuracy
Dropout = 0.2	Regularization	93.7	Minor overfitting observed
Dropout = 0.3	Final model	95.0	Balanced generalization
Without Augmentation	Raw training data	92.8	Overfitting visible
With Augmentation	Final model with augmentation	95.0	Improved generalization

Comparison of the proposed approach with VGG-16 and RESNET-50

The proposed model has a significantly higher training and testing accuracy than both VGG16 and ResNet50. The proposed architecture, which has 24 layers, including 21 convolutional and 3 fully connected layers, achieves 98.23% training accuracy and a testing accuracy of 95.66%, Showing a strong ability to learn and generalization. In contrast, VGG16, which has 13 convolutional and 3 fully connected layers, achieves lower accuracies of 92.02 percent during training and 88.13 percent on testing. ResNet50, even though it has 50 layers, including 49 convolutional layers, shows the weakest performance, with a training accuracy of 91.33% and a notably lower testing accuracy of 79.62%. According to this, the proposed model achieves a better balance between complexity and performance. It avoids the over fitting issues seen in ResNet50 and surpasses VGG16 in learning effectiveness. Overall, the proposed approach demonstrates superior capability for accurate and robust model training. (Table 5) shows the Comparative performance of the proposed VGG model with baseline models VGG16 and ResNet50 using Accuracy, Precision, Recall, F1-score, and the Normalized Leverage Factor ($\gamma\, \omega_m$) (Dey et al., 2022). Higher $\gamma\, \omega_m$ indicates better overall performance (Dey et al., 2023). Normalized Leverage Factor is calculated typically using (15).

$$\gamma\, \omega_m = \frac{1}{n}\sum_{i=0}^{n} \frac{(X_{mi} - X_i^{min})}{(X_i^{max} - X_i^{max})} \tag{15}$$

X_{mi}: Metric value of model m for metric i

X_i^{min}, X_i^{max}: Min and max values for metric i among all models

n: number of metrics used

Table 5 Comparison of proposed approach with VGG 16 and Resnet50

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	$\gamma\, \omega_m$ (Normalized Leverage Factor)	Rank
Proposed VGG	95.66	93.75	94.00	93.75	0.353	1
VGG16	88.13	86.25	86.25	86.25	0.275	2
ResNet50	79.62	80.00	78.00	78.75	0.221	3

A comprehensive evaluation of the proposed approach in comparison to previous approaches

Wheat production globally is greatly affected by rust diseases. Knowing the causes of rust diseases is important for both farmers and researchers to apply the proper methods to stop rust. Rust disease identification by experienced evaluators is a long and challenging process. The issues faced by experienced evaluators have been handled with computer vision methods which now diagnose rust diseases without human help and perform very well. (Table 6) shows the comparison of our approach with previous approaches in this field.

Table 6 Comparison of proposed approach with previous state of art approaches

References	Disease	Techniques	Classification Accuracy
(Kendler et al., 2022)	Yellow rust, stem rust	InceptionResnet V2	94.6
(Genaev et al., 2020)	Leaf rust, stem rust, yellow rust, powdery mildew	EfficientNet B2	94.2
(Kumar et al., 2023)	Wheat Rust	RCNN, Fast RCNN, Faster-RCNN, Mask-RCNN, Resnet-50	93.60
Proposed Approach	Leaf rust, Stem rust, stripe rust	Enhanced VGG	95.66

Conclusion

A new system and special architecture for detecting wheat rust disease is introduced in the study. The model we suggest learns well from training data and does so using a moderate number of resources. A set of real-time images has helped test the system's ability to determine different diseases. The proposed method, with 98.23% accuracy in its training and 95.66% accuracy in tests, is used to distinguish among three kinds of wheat rust diseases (stem rust, leaf rust and stripe rust). The results show that deep learning is now more effective than in the past. When compared to VGG16 and RESNET50, accuracy goes up by 6.21% and 6.90%. Because of this, the methods proposed here can assist in distinguishing wheat rust disease.

Ethics and Consent Statement

All field data used in this study were collected from wheat farms in North Punjab, Pakistan, during February–March 2025. Permission for data collection was obtained from local landowners and relevant agricultural authorities. No identifiable personal data were recorded. The study complied with ethical research practices, and all image data were used solely for scientific purposes, maintaining the privacy and anonymity of field locations and contributors.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

References

Ahmad, A., Saraswat, D., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agricultural Technology*, *3*, 100083.

Chai, Y., Pardey, P. G., Hurley, T. M., Senay, S. D., & Beddow, J. M. (2020). A Probabilistic Bio-Economic Assessment of the Global Consequences of Wheat Leaf Rust. *Phytopathology®*, *110*(12), 1886–1896. https://doi.org/10.1094/PHYTO-02-20-0032-R

Chandio, A. A., Alnafissa, M., Akram, W., Usman, M., & Joyo, M. A. (2023). Examining the impact of farm management practices on wheat production: Does agricultural investment matter? *Heliyon*, *9*(12).

Dey, B., Ahmed, R., Ferdous, J., Haque, M. M. U., Khatun, R., Hasan, F. E., & Uddin, S. N. (2023). Automated plant species identification from the stomata images using deep neural network: A study of selected mangrove and freshwater swamp forest tree species of Bangladesh. *Ecological Informatics*, *75*, 102128.

Dey, B., Haque, M. M. U., Khatun, R., & Ahmed, R. (2022). Comparative performance of four CNN-based deep learning variants in detecting Hispa pest, two fungal diseases, and NPK deficiency symptoms of rice (Oryza sativa). *Computers and Electronics in Agriculture*, *202*, 107340.

Fang, X., Zhen, T., & Li, Z. (2023). Lightweight multiscale CNN model for wheat disease detection. *Applied Sciences*, *13*(9), 5801.

Figueroa, M., Hammond-Kosack, K. E., & Solomon, P. S. (2018). A review of wheat diseases—A field perspective. *Molecular Plant Pathology*, *19*(6), 1523–1536. https://doi.org/10.1111/mpp.12618

Genaev, M., Ekaterina, S., & Afonnikov, D. (2020). Application of neural networks to image recognition of wheat rust diseases. *2020 Cognitive Sciences, Genomics and Bioinformatics (CSGB)*, 40–42. https://ieeexplore.ieee.org/abstract/document/9214703/

Hussain, N., & Maharjan, K. L. (2025). Impact of On-Farm Demonstrations on Technology Adoption, Yield, and Profitability Among Small Farmers of Wheat in Pakistan—An Experimental Study. *Agriculture*, *15*(2), 214.

Jagadeeshan, N., Nagendar, Y., Al-Attabi, K., & AC, R. (2024). Three-dimensional convolutional neural network for wheat rust disease classification. *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, 1–4. https://ieeexplore.ieee.org/abstract/document/10515754/

Kendler, S., Aharoni, R., Young, S., Sela, H., Kis-Papo, T., Fahima, T., & Fishbain, B. (2022). Detection of crop diseases using enhanced variability imagery data and convolutional neural networks. *Computers and Electronics in Agriculture*, *193*, 106732.

Krępski, T., Piasecka, A., Święcicka, M., Kańczurzewska, M., Sawikowska, A., Dmochowska-Boguta, M., Rakoczy-Trojanowska, M., & Matuszkiewicz, M. (2024). Leaf rust (Puccinia recondita f. Sp. Secalis) triggers substantial changes in rye (Secale cereale L.) at the transcriptome and metabolome levels. *BMC Plant Biology*, *24*(1), 107. https://doi.org/10.1186/s12870-024-04726-0

Kumar, D., Kukreja, V., Dogra, A., Goyal, B., & Ali, T. T. (2023). Fusion of Region Extraction and Cross-Entropy SVM Models for Wheat Rust Diseases Classification. *Computers, Materials & Continua*, *77*(2).

Kumar, D., Kukreja, V., & Singh, A. (2024). A novel hybrid segmentation technique for identification of wheat rust diseases. *Multimedia Tools and Applications*, *83*(29), 72221–72251. https://doi.org/10.1007/s11042-024-18463-x

Li, Z., Fang, X., Zhen, T., & Zhu, Y. (2023). Detection of wheat yellow rust disease severity based on improved GhostNetV2. *Applied Sciences*, *13*(17), 9987.

Liu, C. (2020). Classification, machine learning, metrics, precision, recall, ROC-AUC. *KD Nuggets*.

Lubega, J., Figueroa, M., Dodds, P. N., & Kanyuka, K. (2024). Comparative Analysis of the Avirulence Effectors Produced by the Fungal Stem Rust Pathogen of Wheat. *Molecular Plant-Microbe Interactions®*, *37*(3), 171–178. https://doi.org/10.1094/MPMI-10-23-0169-FI

Majumdar, D., Kole, D. K., Chakraborty, A., & Majumder, D. D. (2015). An Integrated Digital Image Analysis System for Detection, Recognition and Diagnosis of Disease in Wheat Leaves. *Proceedings of the Third International Symposium on Women in Computing and Informatics*, 400–405. https://doi.org/10.1145/2791405.2791474

Maulenbay, A., & Rsaliyev, A. (2024). Fungal disease tolerance with a focus on wheat: A review. *Journal of Fungi*, *10*(7), 482.

Nigam, S., Jain, R., Marwaha, S., Arora, A., Haque, M. A., Dheeraj, A., & Singh, V. K. (2023). Deep transfer learning model for disease identification in wheat crop. *Ecological Informatics*, *75*, 102068.

Onyeaka, H., Ejiohuo, O., Taiwo, O. R., Nnaji, N. D., Odeyemi, O. A., Duan, K., Nwaiwu, O., & Odeyemi, O. (2024). The intersection of food security and mental health in the pursuit of sustainable development goals. *Nutrients*, *16*(13), 2036.

Ramadan, S. T. Y., Sakib, T., Al Farid, F., Islam, M. S., Abdullah, J. B., Bhuiyan, M. R., Mansor, S., & Karim, H. B. A. (2024). Improving wheat leaf disease classification: Evaluating augmentation strategies and cnn-based models with limited dataset. *IEEE Access*, *12*, 69853–69874.

Rasheed, S., Tabasam, N., Anwar, J., & Jabeen, S. (2024). Impact of credit on agriculture production in Pakistan: A review. *European Journal of Ecology, Biology and Agriculture*, *1*(1), 3–13.

Ray, R. V. (2024). Effects of pathogens and disease on plant physiology. In *Agrios' Plant Pathology* (pp. 63–92). Elsevier.

Reis, H. C., & Turk, V. (2024). Integrated deep learning and ensemble learning model for deep feature-based wheat disease detection. *Microchemical Journal*, *197*, 109790.

Shahin, A. A., Omara, R. I., Omar, H. A., El-Din, H. S., Sehsah, M. D., Essa, T., Zayton, M. A., & Omar, H. S. (2024). Evaluation of effectiveness resistance genes in wheat genotypes using marker-assisted selection for stripe rust resistance breeding". *BMC Plant Biology*, *24*(1), 785. https://doi.org/10.1186/s12870-024-05470-1

Shahzad, A., Hamid, A., Hussain, A., Rashid, M., Khan, M., & Hussain, Z. (2022). Current situation and future prospects of wheat production in pakistan. *Journal of Life and Social Sciences*, *2022*(1), 5–5.

Tola, F. G., Blitti, K. E. K., Diwan, A., & Mahadeva, R. (2024). Automated Wheat Rust Disease Classification Using Convolutional Neural Networks With Transfer Learning. *2024 IEEE Region 10 Symposium (TENSYMP)*, 1–6. https://doi.org/10.1109/TENSYMP61132.2024.10752155

Verma, S., Kumar, P., & Singh, J. P. (2024). A Unified Lightweight CNN-based Model for Disease Detection and Identification in Corn, Rice, and Wheat. *IETE Journal of Research*, *70*(3), 2481–2492. https://doi.org/10.1080/03772063.2023.2181229

Wani, J. A., Sharma, S., Muzamil, M., Ahmed, S., Sharma, S., & Singh, S. (2022). Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges. *Archives of Computational Methods in Engineering*, *29*(1), 641–677. https://doi.org/10.1007/s11831-021-09588-5

Wheat Plant Diseases. (n.d.). Retrieved 21 May 2025, from https://www.kaggle.com/datasets/kushagra3204/wheat-plant-diseases

Yap, C. K., & Al-Mutairi, K. A. (2024). A conceptual model relationship between Industry 4.0—Food-agriculture nexus and agroecosystem: A literature review and knowledge gaps. *Foods*, *13*(1), 150.