UNDERGRADUATE FINAL YEAR PROJECT REPORT

Department of Computer & Information Systems Engineering
NED University of Engineering and Technology

Deep Learning-Based Rust and Smut Detector for Wheat Plants

Group Number: 04 **Batch: 2020-2021**

Group Members:

Alina Dahani	CS-20073
Muneeba Mubarak	CS-20146
Mohamedamin	CS-17145

Approved by

Dr. Maria Waqas

Assistant Professor

Project Advisor

Author's Declaration

We declare that we are the sole authors of this project. It is the actual copy of the project that was accepted by our advisor(s) including any necessary revisions. We also grant NED University of Engineering and Technology permission to reproduce and distribute electronic or paper copies of this project.

Signature and Date	Signature and Date	Signature and Date
....	....	....

Alina Dahani	Muneeba Mubarak	Mohamedamin
CS- 20073	CS- 20146	CS- 17145
dahanialina@gmail.com	muneebamubarak99@gmail.com	aminmohamed1708@gmail.com
dahani4303786@cloud.neduet.edu.pk	muneeba4309994@cloud.neduet.edu.pk	shire4009969@cloud.neduet.edu.pk

Statement of Contributions

- **All group members** equally contributed on collecting data from all the visits by their own devices
- **All group members** equally participated in Labelling the datasets and in the preprocessing steps
- **All group members** participated in researching about CNN on classification.
- **Mr. Mohammedamin** researched about YOLOv8 model and tested the datasets.
- **Miss. Alina Dahani** researched about InceptionResNetv2 model and tested the datasets.
- **Miss. Muneeba Mubarak** researched about ResNet50 model and tested the datasets.
- **All group members** equally worked in all the models.
- **Mr. Mohammedamin** worked in the interface.
- **Miss. Muneeba Mubarak and Miss. Alina Dahani** worked in report writing.

Executive Summary

The proposed project aims to address the pressing issue of limited access to agricultural knowledge in rural Pakistan, specifically targeting wheat farmers facing the challenges of Rust and Smut diseases. With wheat being the backbone of Pakistan's agriculture, accounting for a significant portion of crop acreage and production, diseases like rusts and smut pose a substantial economic threat, also our project aims to empower farmers through an accessible AI solution. The methodology involves studying diseases, collecting labeled image data from local fields, developing a machine learning module, and creating a user-friendly interface. Key resources include Python, Django, TensorFlow, dedicated GPUs, and collaboration with agricultural experts.

The objective is to develop a Deep Learning-Based Rust and Smut Detector for Wheat Plants, leveraging computer vision and machine learning technologies. The project focuses on studying wheat diseases, collecting data from healthy and diseased plants, and developing a machine learning module capable of accurately detecting Rust and Smut. Additionally, the project aims to create a user-friendly interface tailored to the needs of local farmers, empowering them to independently diagnose and manage these diseases. The significance of the problem lies in the economic impact on food security and stability, emphasizing the need for accessible solutions. The proposed methodology includes a study of wheat diseases, collaboration with agricultural scientists from Sindh Agriculture University in Tando Jam to study wheat plant and its diseases, field visits, and the use of cameras for image data collection. The development of the machine learning module involves data preparation, model selection, training, and evaluation.

This project provides some advantages in agriculture and economy sectors. Firstly, by giving the farmers some long lasting and efficient disease detector to detect and manage the diseases early to avoid crop loss. This helps crops from damaging and providing healthy harvests. Also this early detection makes the farmer to avoid basic financial loss and be stable economically. Our wheat disease detection project increases the food productivity for the growing popularity and promotes sustainable agriculture and economy in Pakistan.

Acknowledgements

This project wouldn’t be a success if it’s not for the guidance and support of individuals who stood by us throughout this project period. We are taking this page as moment to express our heartfelt gratitude to each and every one of them.

First of all we would like to thank our project supervisor Dr.Maria Waqas, Assistant Professor of CISE department, for being a backbone for us and supporting us in every way possible.

Secondly we would like to thank Dr. Ali khan owner of Dahani farms for letting us collaborate with them and guiding us for collecting the wheat data. Also we would like to express our gratitude for the scientist of agriculture university of TandoJam, Mr. Danish for his endless support and Mr. Khalil Ahmed Khanzada, Principal Scientific Officer from university of Karachi for sharing their knowledge with us that helped us bind the agriculture and deep learning together to help our farmers.

We have a special thanks to express to Dr.Khurram, co-chairperson for letting us to collect our data in his own field in Tando Allahyar and guiding us.

Finally thanking our parents, colleagues and friends for helping us to make this project a success.

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
CC	citrus canker
CNN	Convolutional Neural Network
CSS	Cascading Style Sheet
DT	Decision Trees
EID	Emerging infectious diseases
GPU	Graphics Processing Unit
HTML	Hypertext Markup Language
KNN	K-nearest Neighbor
KPK	Khyber Pakhtunkhwa
MLP	Multi-layer perceptron
MNIST	Modified National Institute of Standards and Technology
R-CNN	Region-based Convolutional Neural Networks
SDGs	Sustainable Development Goals
SSD	Single Stage Detectors
STB	Septoria Tritici Blotch
SVM	Support Vector Machines
VGG	Visual Geometry Group
YOLO	You Only Look Once
ResNet	Residual Neural Network
PARC	Pakistan Agriculture Research Council
SARC	Southern Zone Agriculture Research Centre
OpenCV	Open source computer vision library
API	Application Programming Interface
GPU	Graphical Processing Unit
ReLU	Rectified Linear Unit

United Nations Sustainable Development Goals

The Sustainable Development Goals (SDGs) are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace and justice. There is a total of 17 SDGs as mentioned below. Check the appropriate SDGs related to the project.

☐ No Poverty

☑ Zero Hunger

☐ Good Health and well being

☐ Quality Education

☐ Gender Equality

☐ Clean Water and Sanitation

☐ Affordable and Clean Energy

☑ Decent Work and Economic Growth

☑ Industry, Innovation and Infrastructure

☐ Reduced Inequalities

☐ Sustainable Cities and Communities

☐ Responsible consumption and Production

☐ Climate Action

☐ Life Below Water

☐ Life on Land

☐ Peace and Justice and Strong Institutions

☐ Partnerships to Achieve the Goals

Similarity Index Report

Chapter 1 : Introduction

1.1 Motivation and Need

While technology becomes advanced and successful it's still remains out of reach range to many rural side of Pakistan. Even though agriculture is the foundation of Pakistan's economy, farmers face many challenges worldwide.
After finding out a pressing issue: the lack of accessible solutions for two prevalent diseases (Rust and Smut), We are currently working on this project to implement and provide our farmers with user friendly AI tools and resources that will detect these diseases, so that they can independently diagnose, manage, and prevent these diseases. Hence, they can reduce their dependence on external expertise.

The motivation of implementing this AI model is critical challenges faced by the farmers continuously. Since farmers are not notified timely about the crop diseases and often lack access to that information it leads to a rapid loss in yields. Especially Rust and Smut, two potential diseases can cause a big loss if not found and notified in early stages. By implementing a user friendly model, we are currently working on a disease detection application to provide our farmers a best knowledge about the diseases and to protect their crops.

This motive will increase food security and economy stable not only in Pakistan but also worldwide. Detecting the diseases in its early stages will increase the crop yields and agriculture practices.

The need of such a disease detection is, by the fact of growing population of the country, the agriculture sector is becoming more demand to feed the increasing population. It is very important to educate farmers with this technology knowledge and that can help them to overcome the situation of loss in yields.

Therefore, our model will not only detect the disease as early as possible but also provide a better solution sooner. Which paves a prosperous future for the farmers and to the entire agriculture sector.

1.2 Objectives

a) Study Wheat Diseases, Specifically Rust and Smut;

As a primary objective, we are studying diseases that are affecting wheat plants, with a specific focus on Rust and Smut. This involves gaining a comprehensive understanding of wheat plants in general, enabling the identification of both healthy plants and those affected by these diseases, as well as other potential diseases.

b) Collect Wheat Plant Data;

This objective entails the collection of image data for healthy wheat plants, as well as wheat plants affected by Rust and Smut. So far we have collected more than 5000 data altogether and still on process of collecting more data. The collected images will undergo cleaning and preprocessing to prepare them for model training.

c) Development of Disease Detection Machine Learning Module;

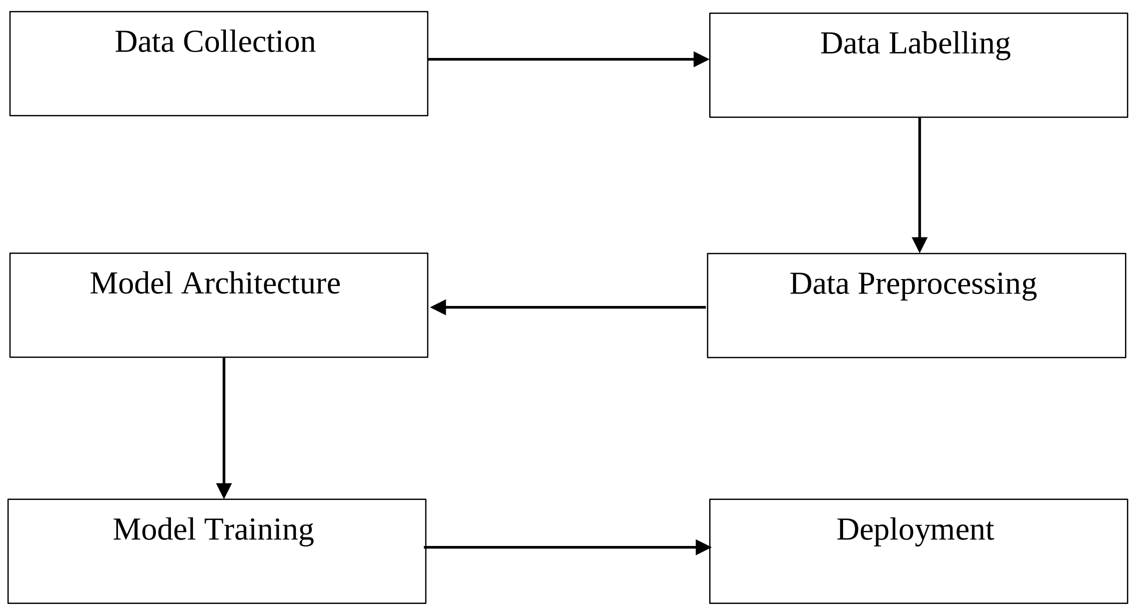
Currently we are working on the creation of a deep learning module capable of distinguishing between healthy and the diseased wheat plants. The module's key function is to accurately recognize Rust and Smut, taking into account their various stages of development.

d) User-Friendly Interface;

This objective aims to design and develop a user-friendly interface tailored to the needs and backgrounds of the farmers we are targeting. The interface should be accessible and easy for farmers to use.

These objectives collectively form the foundation of our project, which seeks to empower farmers in identifying and managing wheat diseases, ultimately contributing to improved crop yields and food security.

1.3 Work Flow



Data Collection: we captured both healthy plant and diseased plants (Rust and Smut) by using our own phone cameras.

Data Labelling: we have labeled each images manually with data types as well as the stages of the diseased plants

Data Preprocessing: we have resized images to a suitable size and fed into the model that can be applicable to all the images.
To get best accuracy we have normalized the pixel size of the images.
To increase the variety of the data or the multiplicity we have augmented them

Model Architecture: learned some models in order to get the best accuracy with transfer learning, Such as YOLOv8, ResNet50 and InceptionResNetv2. Also studied about CNN architecture.

Model Training: Training sets are trained, such as 70% of the dataset, and performances are monitored. Learned the techniques to overcome the overfitting by running many more epochs. Trained with different epochs, batch sizes and learning rates to compare the differences.

Deployment: After the trainings and tests are done and evaluated we have deployed the model to detect the diseases by implementing a user-friendly interface that can be easily accessed by the farmers as well. Our interface is a web application that detects the diseases. The application also finally gives a solution for the disease that are came from scientist’s knowledge

1.4 Tools and Techniques

1.4.1 Tools:

- We are using Samsung model Phone Cameras to capture the healthy and Diseased plants by ourselves and manually Labeling both healthy and Diseases plants.
- We are employing Data preprocessing and training techniques using python libraries such as PyTorch, TensorFlow to enhance the quality of the Data and to highlight the disease symptoms.
- Running the code in Jupyter notebook and google colab.
- Later we will use HTML, CSS, ReactJS for web development □ Other tools: GPU, kaggle kernels, Amazon EC2, Visual Studio

1.4.2 Techniques:

- We are working on statistical Analysis to identify common characteristics between healthy and disease plants.
- We are utilizing python libraries for model development
- We are using pre-trained models as a starting point and fine-tuning them for wheat disease recognition.
- Implementing some techniques to differentiate between healthy, Smut disease and rust disease by using multi class classification.
- Eventually We will use tools for recording user interactions and collecting feedback during testing with the help of testing tools.

1.5 Beneficiaries

Our main target is farmers and the Agriculture field. Detecting the diseases as early and accurate as possible will help to reduce the loss in crop yields.

Improved disease detection will be a beneficial platform for our country's economy factor. When the farmers found out the diseased plants timely and take actions quickly, Consumers would get reliable healthy food for low price.

Quality of the wheat becomes higher, which will increase the agricultural productivity. Project can be a teaching tool to other students, who would like to learn combination studies of both agriculture and computer science.

1.6 Relevance to SDGs

1.6.1 Industry, Innovations and Infrastructure;

1.6.1.1 Industry:

Our project is Directly an advantage for Agriculture industry by detecting the diseases accurately and timely by giving also the solutions to the farmers. This helps to reduce the loss of crop yields and increases economy rate.

Also it paves a way for the healthy and reliable production of food in manufacturing industry and lets public to consume it for affordable prices.

1.6.1.2 Innovation:

Implementing a diseases detection model and interfacing with a user-friendly web application that can be easily accessed by farmers using Deep learning CNN models itself is a significant innovation while traditional methods are time consuming and expensive for the consumers. Our model is well trained by capturing images by ourselves which are valuable and well developed. So the model detects the diseases accurately with high performance.

1.6.1.3 Infrastructure:

Training and testing the dataset, implementing the Deep learning model and deploying the model to the new and fresh data given as input by the user and easy access to the farmers, every step is done perfectly with the help of the infrastructures provided by the cloud computing.

Internet connection is a necessity when operating the model.

1.6.2 Zero Hunger:

By detecting the wheat diseases accurately and early, our project promises a secured food productivity. Also timely detecting the diseases means the amount of the wheat production increases, which highly reduces the hunger.
Farmers are notified by the causing factors and fungicides that they may can get rid of them sooner as possible for the sake of increment of the secured productivity.

1.6.3 Decent work and economy growth:

There will be a stableness in the economy side of the farmers with the help of a model that detects the diseases accurately and timely with best performance.
As we have visited the fields twice we've noticed how farmers manually check the fields for the diseases which consumes more time and money. Our model saves the money from that by implementing a Deep learning model for farmers.

Also our project can help the Agriculture factor to grow higher by providing new jobs in computer system fields as well as in Agriculture field since the diseases management system is an important and reliable practice.

Chapter 2 : Convolution Neural Networks

2.1 Understanding Convolutional Neural Network:

The CNN algorithm is the most well-known and frequently used in the field of deep learning [1]. CNN's primary advantage over its predecessors is that it can identify significant features automatically, without human oversight [2]. CNNs are widely used in many different industries, such as computer vision [3].

To fully use 2D input-data structures, such as picture signals, CNN is used [4]. Images are represented in two-dimensional matrix form, whether they are monochromatic or colored. Each pixel corresponds to a numerical value accordingly. According to that

Convolutional neural networks are a prominent kind of neural network that is specifically made for this use. In recent years, CNN-based architectures have been widely used in computer vision. that significantly enhanced performance [5], due to multiple reasons, such as its versatility in the format of the input data.

A 2D image's spatial details are lost when it is flattened to a 1D vector input. So we need to flatten the image matrix to a 1D vector before feeding an image to an MLP's hidden layers. This entails removing all 2D information from the picture. For 1D signals, treating an input as a simple vector of numbers without any specific structure might be effective, but with 2D photos, this would result in information loss since the network cannot relate the pixel values to one another in order to identify patterns. The fact that these pixel numbers are related to one another and were originally spatially arranged in a grid is unknown to MLPs. Conversely, flattened images are not necessary for CNNs to function.

A CNN network can be fed the raw image matrix of pixels, and it will recognize that pixels near one another have stronger relationships than pixels far from one another [6].

The weight-sharing feature of CNN is the primary factor to take into account. It lowers the quantity of trainable network parameters, which improves generalization and prevents overfitting [6]. The model output becomes highly ordered and very dependent on the extracted features when the feature extraction layers and the classification layer are simultaneously learned [6]. Compared to other neural networks, CNN is far easier to construct on a large scale [4].

2.2 Evolution of CNN Architectures in Image Classification

With the introduction of CNNs, image classification has seen a number of advances and high performance on extensive visual tasks [1]. Deep CNNs (DCNNs) are very successful because of their powerful feature-learning capability [7]. In contrast with traditional image classification techniques, CNN-based classification methods involve an end-to-end learning process wherein the original image serves as the only input, the network performs training and prediction, and the outcome is ultimately the output. This technique breaks the bottleneck of conventional classification methods and does away with the manual extraction of particular visual features. The main benefit of CNNs for picture categorization is also this. In this section, the CNN-based image classification model is primarily introduced, followed by a chronological introduction to each typical classic model [8].

A number of important architecture groups have emerged during the development of convolutional neural network (CNN) models for image classification; these reflect major advances in the field. LeNet, which was built by Lecun et al. in 1998 [9], established the foundation for CNNs by proving the effectiveness of convolutional layers in capturing the spatial hierarchies of features in images. LeNet-5 is the name of the architecture, which consists of three convolutional layers and two fully connected layers, or five weight layers.

In terms of neural networks nowadays, LeNet-5 is a tiny one. More recent networks include millions of parameters, but this one has just 61,706. With the MNIST dataset, LeNet-5 may be trained to achieve accuracy levels above 99%. Because it is a small dataset that is in grayscale and only contains 10 classes.

The 2012 ILSVRC image classification competition was won by AlexNet. The neural network design was developed by Krizhevsky et al. [1] and trained on 1.2 million high-resolution photos to establish 1,000 distinct classes within the ImageNet dataset. Since AlexNet was the first truly "deep" network and made convolutional networks a viable option for the CV community to explore in their applications, it was state of the art at the time.

While AlexNet and LeNet are very similar, AlexNet is larger (more filters per layer) and significantly deeper (more hidden layers). They share the same fundamental elements: a stack of pooling and convolutional layers on top of each other.

Other than a softmax and completely connected layers. As we've seen, AlexNet contains over 60 million parameters and 650,000 neurons, compared to LeNet's approximately 61,000 parameters. This difference allows AlexNet to learn and comprehend more intricate aspects. This made it possible for AlexNet to place highly in the 2012 ILSVRC picture classification competition. AlexNet surpassed all previous opponents by a significant margin. With a top-5 test error rate of 15.3%.

The Oxford University Visual Geometry Group created VGGNet in 2014 [10], hence the moniker VGG. Though VGGNet is a deeper network with more convolutional, pooling, and dense layers, the structural elements are the same as those in LeNet and AlexNet. Apart from that, no new elements are presented here.

There are 16 weight layers in VGGNet, sometimes referred to as VGG16: 3 fully connected layers and 13 convolutional layers. Because of its uniform architecture and ease of understanding, it is well-liked in the DL community. On the ImageNet dataset, VGG16 obtained a top-5 error rate of 8.1% as opposed to 15.3% for AlexNet. Even better, VGG19 managed to attain a top-5 error rate of about 7.4%. Remarkably, despite having more parameters and a deeper pool than AlexNet, VGGNet took fewer epochs to converge because of the implicit regularization that smaller convolutional filter sizes and deeper pools impose [6].

GoogLeNet/inception, proposed by Christian Szegedy et al. in 2014 [11], also had multiple releases later on, such as v2, v3, and v4, and the field advanced quickly. This architecture's primary feature is its ability to create a deeper neural network while making better use of the network's computational resources. The team used a specific version of the Inception network, known as
GoogLeNet, in their 2014 ILSVRC proposal. It achieves far more accurate results while lowering the number of parameters by 12 times (from around 138 million to approximately 13 million) and using a network that is 22 layers deeper than VGGNet. Although the network employed a CNN that was modelled after the traditional networks (AlexNet and VGGNet), it also included a unique component known as the inception module [6].

In 2015, a team from Microsoft Research developed the Residual Neural Network (ResNet). A revolutionary architecture of residual modules with skip connections was introduced by them. Additionally, the network has extensive batch normalization for the buried layers. Compared to smaller networks like VGGNet (19 layers), the team's method allowed them to train very deep neural networks with 50, 101, and 152 weight layers while maintaining a lower level of complexity. In the top-5 error rate category, ResNet managed to attain 3.57% [6].

2.3 CNNs in Object Detection

The object detection task in computer vision refers to the process of identifying the kind and location of various items in an image. In contrast to classification, object detection involves the detection of each and every instance of an object.

So, object detection is essentially a vision task at the instance level. In computer vision, convolutional neural networks (CNNs) have emerged as the state-of-the-art for object detection tasks [6].

There are two distinct categories for state-of-the-art CNN-based object identification models: (a) two-stage detectors and (b) one-stage detectors.
In two-stage detectors such as the R-CNN family, the bounding box is first predicted by the network to have an abjectness score, and it is then put through a classifier to predict the class probability. while in single-stage detectors such as SSD and YOLO, the convolutional layers perform both tasks in a single shot [6] [12].

YOLO is a method that unifies different parts of identified objects into a single neural network. It predicts each bounding box by using the features present in the input image. Each image is divided into a S×S grid, and each grid predicts N bounding boxes as well as a confidence score for these boxes. Whether or not the bounding box encloses the accurate image, the confidence shows how accurate it is. Yolo is incredibly quick and can be used in a real-time setting [13].

2.4 Architecture of Convolutional Neural Networks

Convolutional neural networks (CNNs) are similar to classic artificial neural networks (ANNs). The fundamental component of innumerable artificial neural networks (ANNs) is still a single neuron that receives an input and executes an action (such as a scalar product followed by a nonlinear function). The network as a whole will still express a single perceptive score function (the weight) from the raw picture vectors that are input to the class score that is produced at the end. All of the standard advice and techniques created for conventional ANNs still hold true for the final layer, which will contain loss functions related to the classes [6] [14].

The classification task is not a problem for fully connected layers; they perform it admirably. Our problem stemmed from the way fully connected layers analyzed the picture to extract information.

The biggest limitation of ANN is the computational complexity required for computing image data. For simple datasets such as the MNIST dataset of handwritten digits, which is all in grayscale normalization, ANN can perform quite well, but this computational overhead shows up when we use images that contain more data, such as high-resolution and colored images [14].

In CNN, we kept what was working well in ANN and changed what was not working effectively. Since the completely connected layers aren't extracting features well, let's swap them out with locally connected layers, or convolutional layers. However, given that fully connected layers perform exceptionally well in classifying the retrieved features, we will retain them for the classification phase.

The high level overview of CNN architecture contains these parts.

- Input layer
- Convolutional layers for feature extraction
- Fully connected layers for classification
- Output prediction

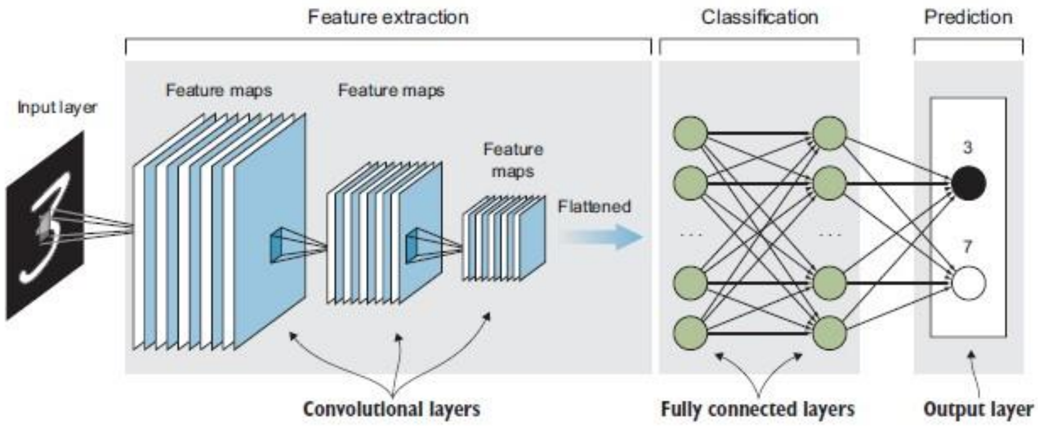


Figure 2. 1 CNN Architecture Overview

Fig. 2.1 The Architecture contains: input layer, convolution layers, fully connected layers and output layer.

Feature extraction is what makes CNN distinct from ANN.

The CNN layers process the image in order to identify patterns and extract features, or feature maps. The neurons learn features from input. A feature map in a CNN is the result of applying one filter to the layer before it. Because it maps out the locations of certain features inside a picture, it is known as a feature map. CNNs search for features like edges, straight lines, and even objects. They add these features to the feature map whenever they find them. Every feature map searches for a different object; for example, one may search for curves, while another may search for straight lines.

After that, the output of this phase is flattened to create a vector representing the image's learnt features [6].

When plotted, nonlinear functions used to have a curvature and are highly significant functions with degrees greater than one. Transforming the input signal into an output signal, which will be used as an input in the subsequent layer, is the primary goal of this layer. Non-linearity layers with popular names include Tanh, ReLU, PReLU, ELU, and logistic or sigmoid layers [13].

The number of parameters that the network must optimize rises with the number of convolutional layers added because this deepens the output layer. As you can see, a large number of parameters (weights) are produced when numerous convolutional layers (often tens or even hundreds) are added. The mathematical operations that occur throughout the learning process become more complicated in terms of both time and space as a result of this growth in network dimensionality. Pooling layers can be useful in this situation. By lowering the number of parameters sent to the subsequent layer, subsampling or pooling aids in shrinking the size of the network, which speeds up the calculation and prevents overfitting in the feature learning layers [13]. In order to lower the total number of parameters sent on, the pooling process resizes its input by applying a summary statistical function, such as a maximum or average. In the CNN design, it is commonly done to add pooling layers after one or two convolutional layers [6].

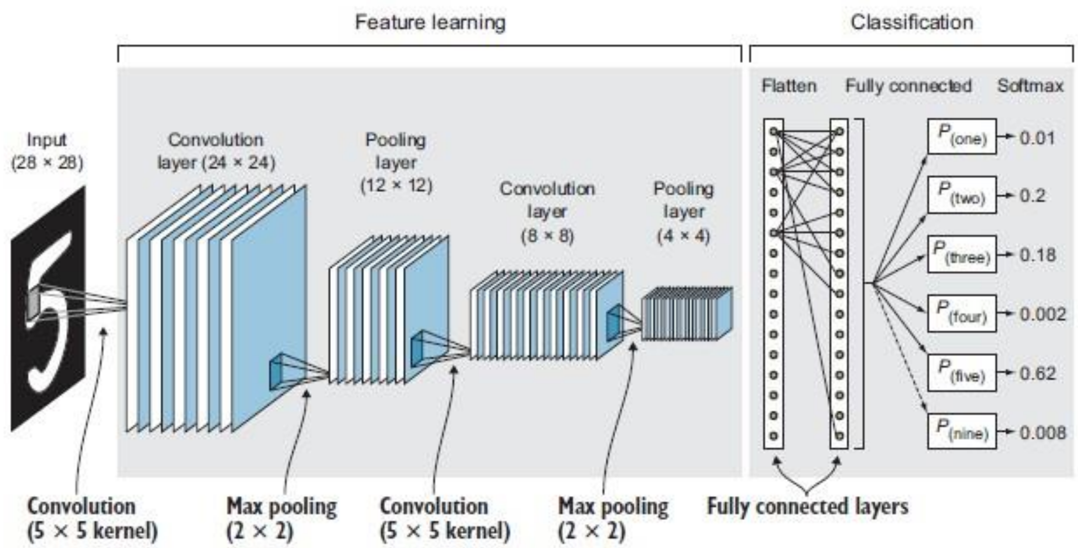


Figure 2. 2 Feature Learning Layer in CNN

Fig. 2.2 The basic components of this part is the Convolution layers and the Pooling layers

We have collected all of the features from the image and placed them in a long tube after putting it through the feature-learning process using convolutional and pooling layers. Utilizing these retrieved attributes to classify images, we will employ the standard fully connected neural network architecture.

2.5 Application of CNNs in Wheat Plant Disease Detection

Wheat is one of the grains that contain important nutrients that our body needs. But due to wheat diseases, the yield decreased. It is usually very difficult to diagnose the disease in its early stages. According to the proposed disease detection models, a two-dimensional CNN model is presented. Which can identify and classify diseases affecting the wheat crop [15], [16]. All these diseases are caused by fungal and bacterial infections. In particular, hoe and rusts affect the whole wheat crop and cause serious losses. To solve this, a convolutional neural network based on deep learning was again approached to study diseases without human involvement. During training, the stochastic gradient descent method achieves high results in classification [17].

Different CNN models like VGG16, VGG19, AlexNet and ResNet are used to train the model for better results. The whole process is divided into five main parts. Data collection, Image preprocessing, Data addition, Training model with different architecture, Result and computation. The system is designed to take an image as input and give a diagnosis result after checking using the model [18]. One of the greatest advantages of CNN is to automatically extract features by processing the raw images captured in-place by camera devices, with various resolutions directly [19].

The use of deep learning convolutional neural networks for the diagnosis of wheat diseases is very useful to improve the quality and quantity of the harvest [20]. Starting from the stages, we can initially use CNN to extract key features between healthy and diseased plants. Second, optimizing their core functions. Then practice the previous qualities. Finally, sending features to CNN for object detection [21]. To amplify small differences between the actual and expected output, we can add as many branches as we need between the deep convolutional layers. It also helps improve adaptability. First, data augmentation techniques are used here to expand the dataset. Second, classifiers such as Softmax, SVM, KNN and Random forest are used to evaluate the data [22].

Chapter 3 : Wheat Diseases

Wheat is the most widely consumed staple food globally, providing essential nutrients like carbohydrates for energy, protein, vitamins, and minerals. It forms the base of diets for billions of people, particularly in temperate zones and developing countries. Wheat cultivation supports millions of farmers and agricultural workers worldwide, contributing to rural economies and food security. It is susceptible to many diseases among them the most common are Rust, Smut, Fusarium head blight (scab), Septoria leaf blotch and Powdery mildew.

3.1 Wheat Rusts:

Rust is a fungal disease and there are three main types of rusts occurring in wheat which are Leaf Rust caused by **Puccinia triticina,** Stem Rust caused by **Puccinia graminis f. sp. tritici** and Strip rust caused by **Puccinia striiformis f. sp. tritici**.

3.1.1 Leaf Rust:

Among all three the leaf Rust or Brown rust is most common [23]. It produces yellow, orange, or brown pustules on the leaves, reducing photosynthetic ability. A unique characteristic is the presence of readily dislodged, rust-colored spores, detectable by gently rubbing the infected leaf surface. It can grow rapidly in the favorable temperature of 10-30 degrees which can cause damage to crops from 10% to 30% in severe conditions [24].

Figure 3. 1 Leaf Rust

3.1.2 Stem Rust:

Unlike round spots, stem rust forms elongated, raised bumps (uredinia). These attack mainly the protective "sheaths" around the stem, but also hit the stem itself, leaves, seed heads, and even tiny "hairs" on them [25]. Unlike many diseases, these bumps usually start underneath leaves, sometimes appearing on top too. It spreads from plant to plant under a favorable temperature of 15-35 degrees damaging the crop from 10% to 70%(severe).

Figure 3. 2 Stem Rust

3.1.3 Strip Rust:

When wheat gets stripe rust, it starts with thin yellow lines showing up on the leaves. Soon after, tiny, bright yellow bumps pop up in neat rows all over the plant, including the leaves, stems, seed heads, and even the little hairs on the seeds. These bumps are full of spores that spread the disease. Higher altitude areas and cold weather favors the growth. Over two-thirds of Pakistan's wheat farms are vulnerable to stripe rust. Provinces like Khyber Pakhtunkhwa, Punjab, and Baluchistan have witnessed a gradual uptick in the presence of stripe rust within their borders [26].

Figure 3. 3 Stripe Rust

3.2 Wheat Smut:

Smut is also one of the fungal diseases in plants and in wheat five bunt and Smut diseases are associated, Loose Smut, Flag Smut. Among wheat diseases, common bunt (stinking smut), dwarf bunt, and Karnal bunt reign supreme after rust, particularly thriving in the Near East due to their perfect adaptation to the region's environmental conditions**.** Both diseases exhibit a preference for wheat seeds, impacting their quality and quantity, with subsequent negative consequences for overall crop health and yield.

3.2.1 Loose Smut:

Loose smut of wheat is caused by Ustilago segetum (Pers.) Rostr. var. tritici occurs throughout the wheat growing regions of the world. This fungal pathogen causing loose smut of wheat infects the seed embryo and remains dormant until the plant grows, at which point it spreads throughout the entire plant systemically. The disease can occur anywhere wheat is grown, but it's less prevalent in warmer regions where the wheat plant completes its life cycle more quickly. Before wheat heads emerge, some plants might show faint yellow streaks and stiff, dark green leaves. Once the heads appear, they quickly release powdery spores that infect other wheat plants during flowering, leading to new seed infections [27].

Figure 3. 4 Loose Smut

3.2.2 Flag Smut:

It is caused by **Urocystis tritici/Tilletia tritici.** The presence of this disease is indicated by white to yellow streaks developing on infected leaves and their sheaths. These streaks undergo a color change, becoming gray and subsequently black. The infection results in the formation of long, thin, black pustules (blisters) specifically located between the veins on leaves. Initially covered by the plant's outer layer, these pustules eventually burst open, releasing black spores and causing the leaves to tear into ribbon-like fragments. Flag smut is a disease found in various regions of Pakistan including Punjab, Baluchistan and KPK [28].

Figure 3. 5 Flag Smut

3.3 Common bunt (Stinking bunt):

Common bunt, caused by fungal pathogens Tilletia caries and T. leaves, reduces both the quantity and quality of wheat harvests. The infected wheat's kernels appear plump and dark lacking the nutritional value and giving stinking odor due to the presence of fungal spores. The higher regions of Baluchistan, KPK, and Murree hills in Pakistan are hotspots for common bunt, posing a particular threat to locally cultivated wheat varieties. A rock blight outbreak can be catastrophic, decimating 70% of grain ears. This translates to unsellable crops for farmers and long-term soil contamination, jeopardizing future harvests.

Figure 3. 6 Stinking Bunt

Similar to common bunt, but distinct in odor and impact, **dwarf bunt** emerges from a different fungal culprit (Tilletia controversa) and attacks spring wheat in snowy regions. Notably, this disease avoids the foul smell associated with common bunt.

Karnal bunt, attributed to the fungus Tilletia indica (also known as Neovosia indica), along with black point, caused by fungi such as Alternaria spp, Helminthosporium, and Curvularia, are the primary seed-borne diseases affecting wheat.

3.4 Fusarium Head Blight (SCAB):

Head blight, also known as scab, significantly reduces wheat yield and raises concerns for human and animal health due to the production of mycotoxins [29]. The disease, caused by several Fusarium species including Fusarium graminearum, starts by bleaching individual spikelets and then moves upwards and downwards within the wheat head, sequentially infecting additional grains. It occurrence depends on the weather conditions, during the flowering stage, rainy weather accompanied by high temperatures and ample primary inoculum can lead to significant yield losses, with potential reductions of up to 80% in crop yield.

Figure 3. 7 Fusarium Head Blight

3.5 Septoria Tritici Blotch (STB):

Septoria tritici blotch (STB), induced by the ascomycete fungus Mycosphaerella graminicola (also known as Septoria tritici in its asexual stage), stands out as a critical foliar ailment affecting wheat.

This disease is distinguished by the formation of necrotic lesions on both leaves and stems, which emerge subsequent to the collapse of infected cells. It tends to be more widespread during periods of cool, damp weather.

Blotches on the leaves are a result of early ascosporic infections. These blotches harbor another type of fruiting body called pycnidia within them.

Figure 3. 8 Septoria Tritici Blotch

3.6 Powdery Mildew:

Powdery mildew, a fungal disease affecting wheat crops, emerges from various Blumeria species. Blumeria graminis f. sp. tritici stands as the most prevalent culprit, posing a significant threat to crop yield and quality if proper management strategies are not implemented.

The initial signs of infection are white, powdery dots, typically circular in shape, appearing on leaves, stems, and heads of plants. These lesions gradually expand and merge, forming larger, continuous areas covered in a powdery substance.

It preferentially develops and proliferates in dry and warm conditions, with a specific temperature range of 59 to 77°F (15-25°C).

Pakistan saw varying degrees of powdery mildew across regions and years. While Sindh had no cases and Punjab minimal issues, the disease hit harder in Khyber Pakhtunkhwa (17% fields) and Kashmir (11% fields) in 2018 and 2017. Gilgit-Baltistan had few isolated cases. Overall, the north is most susceptible, requiring focused management efforts to protect yields [30].

Figure 3. 9 Powdery Mildew

Chapter 4 : Literature View

4.1 Plant diseases all over the world

In recent years, plant diseases caused by bacterial, fungal and viral infections have increased worldwide. Which greatly affects performance and economic efficiency. Despite the fact that plants have innate immune cell immunity, plant pathogens have the ability to evade this immunity [31], [32]. One of these pathogens have developed new infectious diseases (EID) that are stronger in the agricultural community, causing a major pandemic [33]. The main causes of this pandemic are the spread of spores, which are mostly spread by nature [34]. When it comes to weather problems, climate change bears the greatest responsibility of the agricultural industry. Climate change will affect the reproduction and growth of these pathogens, ultimately leading to a global threat [35], [36].

According to studies, improper and overdose storage of chemicals is also an indirect cause of these plant diseases. which lead to global crop losses hence, the poor countries were forced to face poverty and to pay more for the food they needed from the market [37]. To protect the plant against these diseases, one must follow some intelligent steps and methods in observing, recording and detecting them. Which can be done both scientifically using sensors and technologically applying different algorithms [38]. Disasters often occur because pathogens are often used in agricultural models and undergo most chemical poisons and treatments. Now scientists are working to find a refuge for these plants to protect themselves from this tragedy [39].

4.2 Worldwide Wheat Diseases

One of the main foods or basic cereals consumed worldwide is wheat. The increasing global population necessitates a linear relationship between dietary changes and wheat demand. which is entirely dependent on treating bacterial or fungal-infected illnesses

[40], [41], . The wheat plant community is under risk due to the global movement of the majority of harmful fungus. The four wheat diseases that cause the most worry are leaf spots, wheat blast, root infections, and head blight [42]. Diseases have spread widely, even in regions where seeding is improperly done or produces a low proportion of output. Fungicide use can, to some extent, stop this from happening [43].

Stripe rust is the most prevalent wheat disease, according to studies, and it can occur anywhere there is wheat agriculture worldwide. It is also known as yellow rust and typically appears in damp, chilly climates [44]. Another illness known as Take-all, which gained notoriety in South Australia, is likewise transmitted by moist soil. Rotten roots, clogged roots, and yellowing plants were signs of this disease. For ten years, the illness had the greatest impact on crop growth [45]. The three rust and powdery mildew fungus have historically severely damaged crops even though these diseases reduce wheat production [46], [47] .

Many studies have been conducted to control this disease of wheat in the United States. The abbreviation MoreCrop is used to describe this wheat disease management. According to the user's information about the environmental change, the management only gives the process for the user control for the disease [18]. As a chemical treatment, foliar complex and some fungicides have been accepted by growers around the world to control tan spot [48].

4.3 Plant and Wheat Diseases in Pakistan

Pakistan, being predominantly reliant on agriculture, faces challenges similar to many other regions worldwide, contending with a myriad of bacterial, viral, and fungal diseases affecting a diverse range of crops, including fruits, vegetables, grains, and pulses. Consequently, this has led to a decline in crop yields

4.3.1 Bacterial Diseases

Citrus Canker (CC) has caused concern for the sustainability of citrus production worldwide.

Pakistan's citrus yields lag behind global leaders, averaging 11 t/ha compared to China's 27, Brazil's 26, and India's 22 t/ha [49] resulting in contribution of only 2% in the world citrus production [50].

Moreover, bacterial wilt, attributed to Ralstonia solanacearum, poses a grave risk to the cultivation of chili peppers in Pakistan [51]. The pathogen is pervasive throughout the country, with its prevalence and distribution extensively researched in various agro ecological zones and districts.

Studies reveal that the disease is predominantly concentrated in Islamabad Capital Territory and Punjab, exhibiting incidence rates of 19.2% and 13.9%, respectively. Ralstonia solanacearum is recognized as the world's second most destructive bacterial phytopathogen, impacting a broad spectrum of plant species. This ailment significantly hampers chili production in Pakistan [52].

4.3.2 Fungal Diseases

Wheat rust, smut, and powdery mildew are fungal diseases that affect various crops like wheat, barley, and sunflower [53]. Rust causes reddish-brown pustules on leaves and stems, smut causes distorted and blackened kernels, and powdery mildew covers leaves with a white powdery growth, hindering photosynthesis [43]. In Pakistan, wheat rust pathogens, including yellow rust and stripe rust, are highly diverse and recombinant. Genetic recombination has been detected within P2 isolates, and resistance genes have been postulated in 40 Pakistani wheat cultivars [54]. The high diversity of Puccinia striiformis in Pakistan represents a potential threat to wheat production and can production rate up to 30% and destroy the crop within a month after its first assault, in the region and as a possible source to found clonal populations. The lack of clear population subdivision could be attributed to migration of the pathogen [55].

The timely identification of diseases is crucial to minimize later-stage yield losses. Presently, disease detection relies on manual inspection, a method that is both time-consuming and demands substantial human resources and expertise. Moreover, it is influenced by the experience and perspective of the individuals conducting the inspection. To address these challenges, automated disease detection methods employing image processing and machine learning techniques have been adopted [56]. Modern generic approach has been proposed for the identification and classification of wheat diseases using Decision Trees (DT) and different deep learning models, with an accuracy of 28.5% and 97.2% respectively [57].

Recently, the dataset for wheat rust is locally sourced from the National Agricultural Research Centre in Islamabad. A pre-existing U2 Net model is employed to eliminate the background and isolate the leaves with rust disease. Following this, two deep learning classifiers, namely the

Xception model and ResNet-50, are utilized to categorize the severity levels of stripe rust. The ResNet-50 model demonstrated superior performance, achieving the highest accuracy at 96% [58].

Chapter 5 : Data Collection

5.1 Introduction:

Securing a solid dataset is a crucial challenge when constructing machine learning models for agricultural research. While having 5,000 separate photos of wheat fields from Sindh looks large, ensuring that the collection adequately depicts the region's various agricultural contexts poses a significant difficulty. To cover the wide range of healthy and sick plant states in varied environmental situations, thorough research and techniques are required. A substantial and carefully curated dataset is required for training algorithms to properly identify illnesses in real-world circumstances.

5.2 Objectives

- **Primary Goal:** The project's main objective is to collect a valuable dataset of 1000 images containing healthy, Stem Rust, Stripe Rust and Smut to train a reliable machine-learning model that can correctly recognize and categorize plant diseases from photos. This will enable farmers to protect their crops and boost productivity by acting quickly.
- **Diversity:** To incorporate photos of different wheat plant species in a range of environmental settings, showing both healthy and diseased plants.
- **Volume**: To guarantee reliable model testing and training, assemble a sizable dataset of at least 10,000 photos.
- **Quality:** To obtain high-resolution photos that show the minute details required to differentiate between symptoms of comparable diseases.

5.3 Data Sources:

1. **Farm Locations:** To guarantee a diverse dataset, farms were chosen according to their diversity in crop types, geographic areas, and farming techniques. Among them were farms with a track record of particular plant diseases produced artificially.

2. **Geographic Distribution:** To get a broad range of climatic circumstances and plant health statuses, data was gathered from 4 different farms spread throughout three regions (Tando Allahyar, Usman Shah Huri, Tando Jam and Karachi).
3. **Plant Varieties:** The image dataset includes images of 80 different varieties of wheat such as Benazeer, Morocco, Sindhu, and Imdad in national trials.
4. **Environmental Conditions Considered:** Data was collected under various weather circumstances (sunny and cloudy) to determine how these factors influence the emergence of illness symptoms. Images were captured at several times of day (morning, afternoon, and evening) to allow for fluctuations in natural lighting.

5.4 Data collection methods:

Specific photographic techniques were used to capture precise and clear pictures of plants and their settings, ensuring high-quality and consistent photographs for the wheat dataset. Three cameras of the Samsung A series were used with default camera settings.

5.5 Data Collection Procedures:

5.5.1 Planning and Preparation:

Reached out to Mr. Danish, an Agricultural Data Scientist at Dahani Farms, as well as Prof. Mohammed Khurram and Khalil Khanzada from Sindh Agricultural University and PARCO Karachi University respectively, to obtain permission for field visits. Coordination for visit scheduling was conducted, with six separate visits arranged based on the availability of the respective owners or representatives of the fields.

5.5.2 Details of Visit:

Visit 1: Barley & Wheat Research Institute

Date: Dec 29th, 2023

Location: Sindh Agricultural University, Tando Jam

Objective: The first survey to see the wheat crop and observe different varieties at the growing age.

Activities:

- **Arrival and Setup**: Arrived at the field at 11:00 AM. Visited and explored different varieties of growing wheat.
- **Briefing and guidance:** Mr. Danish made us visit the entire research land of 42 acres, having different varieties of wheat, durum and barley.

Observations:

- The wheat was at a very early stage and there were rare/no traces of any of the diseases.
- The field was having many varieties under supervision to find the most resistant variety in the province.
- Benazeer wheat was the most resistant variety under consideration and Morocco was highly susceptible to disease as per the previous data collected.
- Some of the varieties were grown a week earlier to see the effect of the atmosphere.
- Scientist predicted that the weather would be mild and favorable for the wheat so it won't be affected much by the diseases this time.

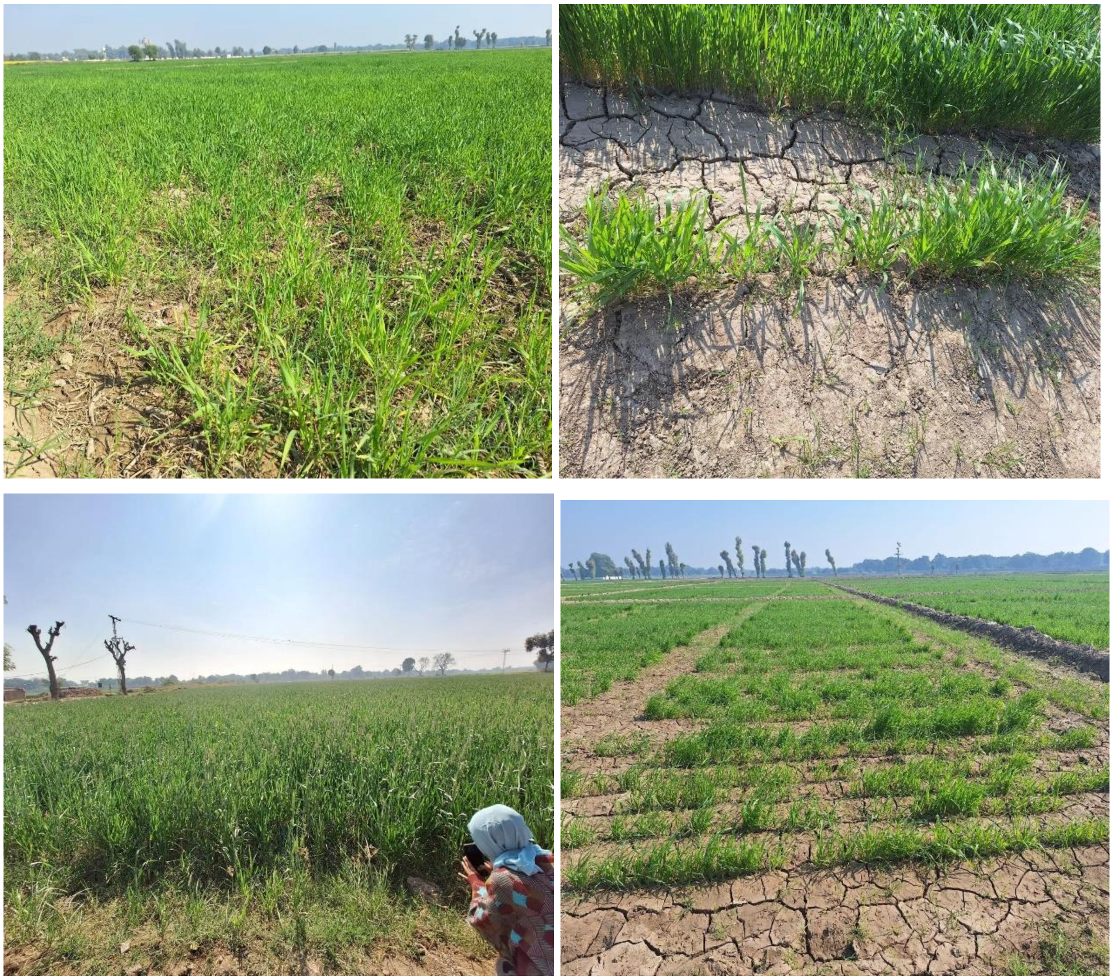


Figure 5. 1 Visit 1 Barley & Wheat Research Institute

Visit 2: Mir Talpur's Field under Prof. Dr.Khurram

Date: Dec 30th, 2023

Location: Usman Shah Huri, Tando Allahyar

Objective: The first survey to see the wheat crop and observe it as an early age of 4 weeks.

Activities:

- **Arrival and Setup**: Arrived at field at 12:00 Pm. Visited and explored different varieties of growing wheat.
- **Briefing and guidance:** Prof. Dr. Khurram made us visit his field having different varieties of wheat.

- **Image & Video Capture**: Collected plenty of healthy wheat pictures at initial stage.

Observations:

- The wheat was in its early stages, with little or no signs of illness. However, the soil lacked nutrients, causing some the plant leaves to become yellow.
- The field was having mix of many varieties considered resistant in the market.
- The field had sufficient water to support optimum development.

Figure 5. 2 visit 2 Mir Talpur's Field under Prof. Dr.Khurram

Visit 3: Dahani Field under Dr. Ali Khan Dahani

Date: Dec 30th, 2023

Location: Usman Shah Huri, Tando Allahyar

Objective: The first survey to see the wheat crop and observe it as an early age of 6 weeks.

Activities:

- **Arrival and Setup**: Arrived at field at 3:00 Pm. Visited and explored different varieties of growing wheat.

- **Briefing and guidance:** Dr. Ali Khan made us visit his field having different varieties of wheat.
- **Image & Video Capture**: Collected plenty of healthy wheat pictures at the initial stage.

Observations:

- The wheat was in its middle age, it was healthy overall and there were no signs of any disease. It was sprayed with pesticides and fertilizers to keep the crop healthy
- The field was having combination of many resistant varieties such as Akbar-19,Satar, Sindhu and Seher.

Figure 5. 3 visit 3- Dahani Field under Dr. Ali Khan Dahani

Visit 4: Barley & Wheat Research Institute

Date: Feb 12th, 2024

Location: Sindh Agricultural University, Tando Jam

Objective: The second survey is to see the wheat crop and observe different varieties in middle age with infections.

Activities:

- **Arrival and Setup**: Arrived at the field at 11:00 AM. The field was surveyed for infections.

Observations:

- The wheat was at the middle stage and there were some traces of any of Rust and Smut but no traces of Strip rust because of the favorable weather
- Benazeer wheat was the most resistant variety and no traces of any disease were found while Morocco had most of the Rust.
- Smut was observed in both wheat and barley.

Figure 5. 4 visit 4- Barley & Wheat Research Institute

Visit 5 : **PARC-SARC-Crop Diseases Research Institute Karachi**

Date: March 7th, 2024

Location: University of Karachi, Sindh.

Objective: The first survey to see the wheat crop artificially infected by brown and brown rust.

Activities:

- **Arrival and Setup**: Arrived at field at 12:00 Pm. Visited and explored affected wheat.
- **Briefing and guidance:** Mr. Khalil Ahmed Khanzada Principal Scientific officer (PSO) gave brief session on how they artificially infect plants with diseases.
- **Image & Video Capture**: Plenty of diseased plant pictures were captured.

Observations:

- The wheat was in its middle age, it was covered in rust mostly. Other plants were getting affected by that due to wind.
- The field was having an area of 2 acres with myriad varieties in small portions.

Figure 5. 5 visit 5- PARC-SARC-Crop Diseases Research Institute Karachi

Visit 6: PARC-SARC-Crop Diseases Research Institute Karachi

Date: March 8th, 2024.

Location: University of Karachi, Sindh.

Objective: The second survey was to collect the most pictures of Loose Smut.

Activities:

- **Arrival and Setup**: Arrived at the field at 10 AM
- **Image & Video Capture**: A multitude of images featuring Smut and brown Rust were captured.

Observations:

- This survey was targeted to observe the wheat infected with Loose Smut.
- There were not many traces of Smut because of weather being favorable for the crop.

Figure 5. 6 visit 6-PARC-SARC-Crop Diseases Research Institute Karachi

Figure 5. 7 Data Collection of 2 visits

During the 5th and 6th visits, a comprehensive dataset comprising 1,650 images of Stem Rust was collected.

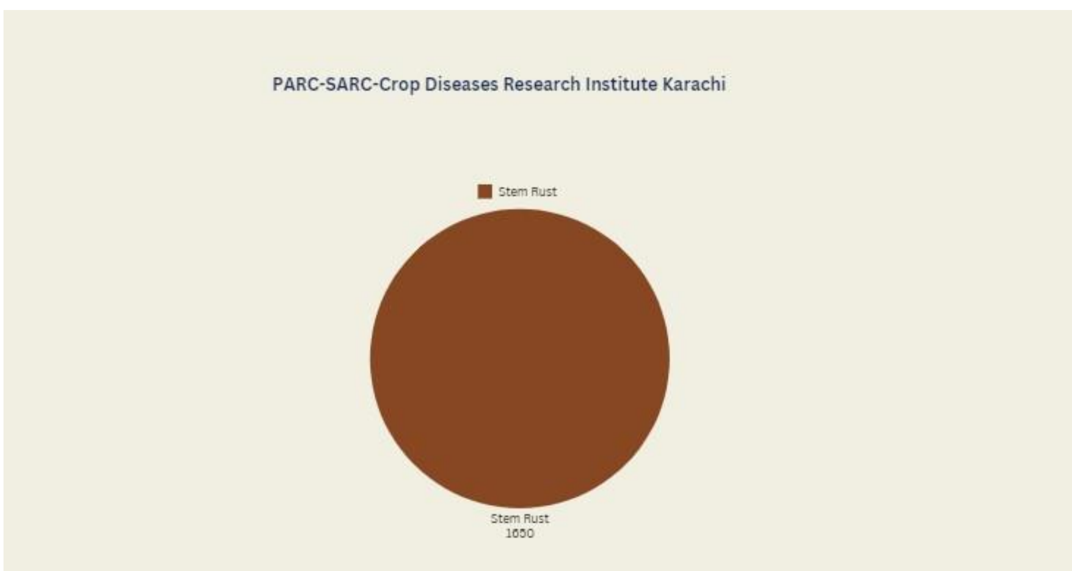


Figure 5. 8 Data Collection third visit

Around 5,000 high-quality images of healthy wheat plants were meticulously extracted from a pool of 75 videos. This extensive dataset represents various growth stages and environmental conditions of wheat, serving as a valuable reference for comparative analysis with diseased samples. In total, the project amassed a dataset of 10,000 images.

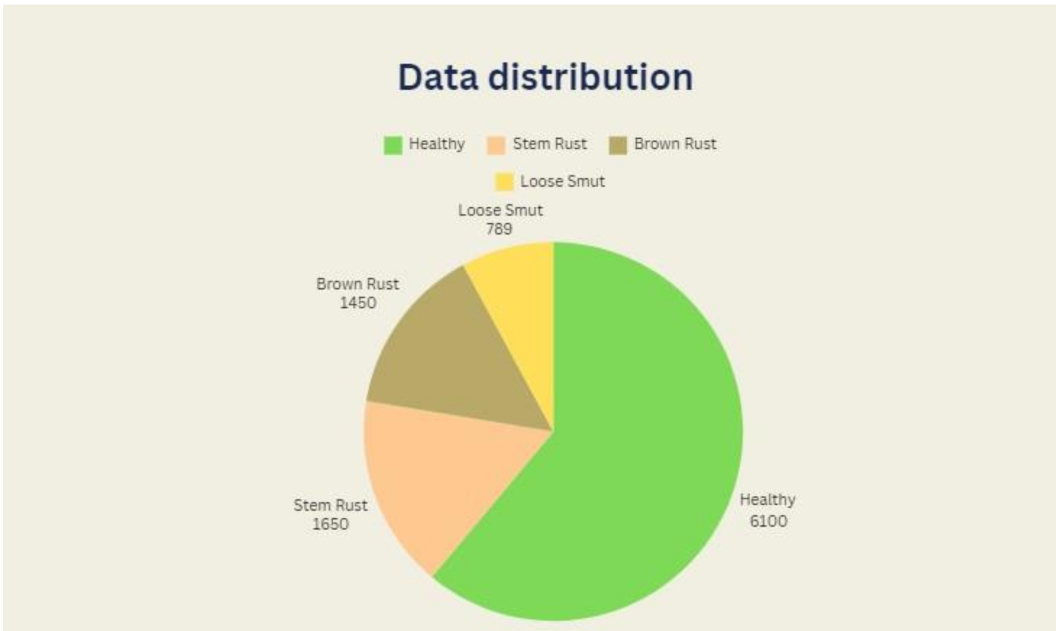


Figure 5. 9 Data Distribution

5.6 Data Preprocessing:

5.6.1 Data Cleaning:

The images were captured with careful consideration of the prevailing conditions. The overall quality of the dataset was good; however, a few images were found to be blurry and noisy. All such subpar images were promptly discarded to maintain the dataset's Integrity

5.6.2 Data Reduction:

Dimensionality Reduction: To standardize the dimensions of the images, we utilized the `cv2.resize` function from OpenCV. Each image was resized to approximately 640x840 pixels.

Different scaling factors, `fx` and `fy`, were applied to various images to achieve this consistent size. This approach ensured that all images had uniform dimensions, which was crucial for subsequent processing and analysis tasks.

5.6.3 Data Splitting:

- **Partitioning**: The dataset was partitioned into separate subsets for training, testing and purposes by using the 70/20/10 rule. This ensures an unbiased evaluation of model performance and prevents overfitting.

- **Stratification**: Classification was conducted among various diseases affecting wheat plants. Stratification was employed to maintain the proportional distribution of each disease class across all subsets, thereby enhancing the reliability and validity of the classification results.

5.6.4 Data augmentation:

Table 5. 1 Data Augmentation

Model	Class	Before Augmentation	After Augmentation
	Healthy	5000	5000
Healthy Vs. Diseased	Diseased	3562	5000
	Rust	2984	2984
Rust Vs. Smut	Smut	578	2910
	Brown Rust	1356	1628
Brown Rust Vs. Stem Rust	Stem Rust	1628	1628
	Low	431	431
Brown Rust scoring	Medium	421	421
	High	240	484
	Low	343	1028
Stem Rust Scoring	Medium	877	1027
	High	418	1036
	Low	43	209
Loose Smut Scoring	Medium	235	235
	High	216	216

- Above table shows the Data augmentation done to balance the dataset using various augmentation techniques. These included Gaussian blur, flipping, and rotation. Gaussian blur added noise and varied image clarity while flipping and rotation created mirrored and rotated versions of the images. By implementing these techniques, the dataset's imbalance was addressed, enhancing the model's performance in disease diagnosis.





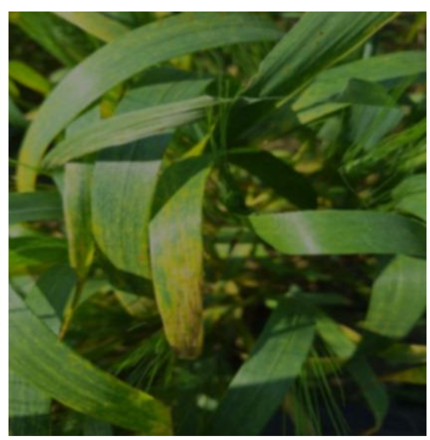



Figure 6. 1 Augmented Data

5.7 Data Organization:

Organizing image datasets into coherent categories is essential for efficient data management and retrieval. It was the most time-consuming and time-consuming task. Here's how the data was structured and organized:

1. **Categorization into Folders**: Images were categorized into different folders based on relevant criteria such as plant disease type.
    - ○**Example**: Images of wheat plants affected by various diseases were grouped into separate folders for each disease type (e.g., "Healthy", "Stem_rust", “loose_Smut”,"brown rust").

By implementing a structured approach to image storage, backup, and organisation, we ensured the accessibility, integrity, and usability of our image dataset for subsequent analysis and modelling tasks.

5.8 Data Management:

Image Storage and Backup

To ensure the integrity and accessibility of image dataset, a robust storage and backup system was implemented. Here are the key components:

1. **Directory Structure**: A hierarchical directory structure to organize the images systematically. Each level of the structure serves a specific purpose, facilitating efficient retrieval and management of the dataset.
    - **Root Directory**: The top-level directory serves as the main repository for all image data.
    - **Subdirectories**: Within the root directory, subdirectories are organized based on different criteria such as plant species, disease type. This hierarchical organization helps maintain clarity and orderliness.

2. **File Naming Conventions**: Adopting standardized file naming conventions is crucial for easy identification and retrieval of images. A consistent naming scheme that incorporates relevant information such as disease type was incorporated.
    - **Example**: For instance, an image of a wheat plant affected by brown rust captured were named ,"Healthy_1.jpg",” Stem_rust_1.jpg etc.

3. **Backup Strategy**: Regular backups of the entire image dataset were performed to safeguard against data loss or corruption. Backup copies were stored in Personal computers as well on the cloud to mitigate risks associated with hardware failures or disasters.

5.9 Challenges and Limitations:

- **Lack of access to fields:** Planning and travelling for each visit involved communicating with university authorities to obtain permission to travel outside the challenging city, as well as coordinating with the fields to be visited. Initially, it was necessary to wait for the wheat plants to grow after sowing. Following this, coordination with different authorities was required to find a suitable window for collecting diseased plant samples and obtaining guidance from experts. During the week when the first diseased dataset was collected, it rained, which washed away most of the disease symptoms from the plants. This added an extra burden, making it more challenging to find sufficient diseased data.

- **Images which represent all:** The team emphasized the inclusion of images taken from various perspectives. Acknowledging their limited size and potential bias, consideration
  was given to how end users and individuals using low-resolution cameras might capture images. This approach aimed to ensure the dataset's representation of real-world scenarios, enhancing the analysis and models' robustness and applicability.

- **Robust and well-balanced dataset**: There are natural elements such as lightning, time of the day and the angles which the picture was taken from, as well as elements related to the growing scheme or the scheme on which the disease might be founded such as border lines which are plants sowed in the outer corners of the each square, these plants will have more soil in the background and their appearance will be different due to direct weather exposure compared to the wheat grown in the mid which will have other wheat in its surrounded background. We tried to maximize this diversity by collecting images for all of these varying conditions. So we have to go back to the same fields at different times of the day and different places in the field.

- **Climatical limitation:** As the fields which we covered are located in Sindh province, there are climatical constrains to that, due to Sindh weather being relatively warmer than other parts of Pakistan in the wheat growing seasons, there are only a few types of diseases which effect the wheat which is grown in the province. This led us to search for more resources while still maintaining the integrity of our data, we found wheat nursing fields from Pakistan Agriculture and Research Crops and were able to collect images of diseases which only effect the wheat plants in colder parts of Pakistan such as Punjab and KPK.

- **Videos Challenge:** Video footage of wheat was collected to obtain images from various angles, which are challenging to capture in pictures. However, issues arose when converting the videos to images. Despite extracting 8,000 images initially, after screening the relevance of each image, the number of useful images from the videos was reduced to 5,000.

- **Aspect ratio:** We have tried resizing images while making inferences this was difficult to generalize into any image that might come into the model for inference, some photos will come in relatively small dimensions which will need to scale up rather than scale down. The best inference was found when images were approximated to the nearest multiple of 32, this number 32 is because the stride length in the backbone of YOLO V8 is 2^5, this is the final implementation approach we used in our inference.

Table 5. 2 Summarized Dataset count

Model	Class	Total Count	Split Count		
			Train 70%	Val 20%	Test 10%
Healthy Vs. Diseased	Healthy	5000	3500	1000	500
	Diseased	5000	3500	1000	500
Rust Vs. Smut	Rust	2984	2089	597	298
	Smut	2910	2037	582	291
Brown Rust Vs. Stem Rust	Brown Rust	1628	1140	326	162
	Stem Rust	1628	1140	326	162
Brown Rust scoring	Low	431	302	86	43
	Medium	421	295	84	42
	High	484	339	97	48
Stem Rust Scoring	Low	1028	720	206	102
	Medium	1027	719	205	103
	High	1036	725	207	104
Loose Smut Scoring	Low	209	146	42	21
	Medium	235	165	47	23
	High	216	151	43	22

Chapter 6 : Models

6.1 Model YOLOV8

Yolov8 is the new computer vision model in the Yolo series built by ultralytics which supports object detection, classification and segmentation tasks. Yolov8 is easily accessible through python libraries as well as command line interfaces. It has improvement in both architectural and developer experiences. Key features and optimizations of this Yolov8 make it an ideal choice for various object detection tasks in a wide range of applications, offering cutting edge performance in terms of accuracy and speed.

Although yolov8 is basically famous for object detection, without any doubt they are equally adaptable for classification tasks as well. Object detection is basically identifying different objects within an image, whereas classification does categorizing entire image into already defined classes. In classification Yolov8 focuses on the features which are extracted from the image carefully and apply them to classify the images that are fed into the model.

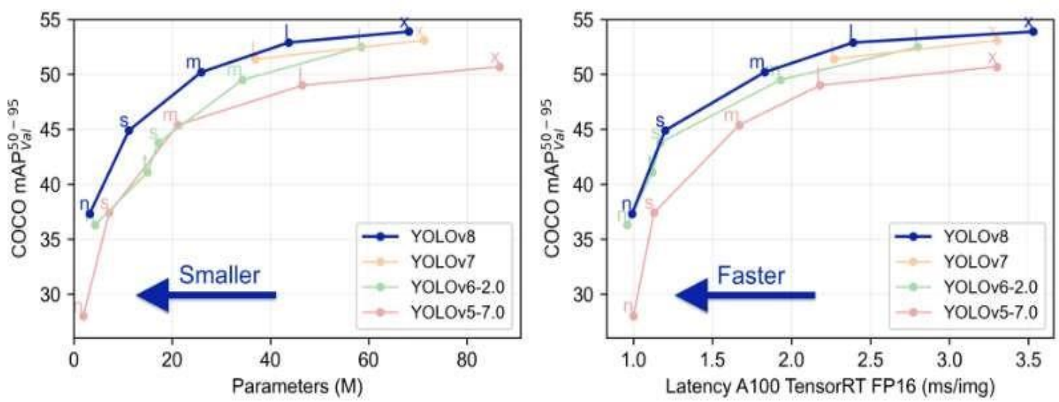


Figure 6. 2 Performance of YOLOv8

6.1.1 Yolov8 Architecture:

Key features of Yolov8 are, it employs state-of-the-art backbone and neck architectures, resulting in improved feature extraction, object detection and classification performance. Secondly it adopts an anchor-free split ultralytics head which results in best accuracy and detection process, to capture the most features convolution neural networks with its fully connected layers map those extracted feature to the classification categories. This way the architecture is very useful to differentiate even a narrow difference between the similar classes.

Yolov8 provides a variety of pre-trained models as per different task requirements Variations of Yolov8 (Eg: yolov8nano, yolov8small, yolov8m, etc.) highlights its compatibility with various operational modes such as inference, validation, training and export. This shows the versatility and robustness of the Yolov8 series.

Model	size (pixels)	acc top1	acc top5	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B) at 640
YOLOv8n-cls	224	69.0	88.3	12.9	0.31	2.7	4.3
YOLOv8s-cls	224	73.8	91.7	23.4	0.35	6.4	13.5
YOLOv8m-cls	224	76.8	93.5	85.4	0.62	17.0	42.7
YOLOv8l-cls	224	76.8	93.5	163.0	0.87	37.5	99.7
YOLOv8x-cls	224	79.0	94.6	232.0	1.01	57.4	154.8

Figure 6. 3 Pre-trained models of Yolov8

6.1.2 Ease of interface in yolov8:

Yolov8 is basically provides a user-friendly interface where you can simply install it by using, !pip command. This allows us to initiate the working very easily without excessive configurations, while we can adjust all the model parameters for training, validation and prediction. Since yolov8 is a pre-trained model it helps the model for quick transfer learning which adapts itself quickly for our custom dataset.

Yolov8 also provides a brief and clear API for training, validation and inference. Function are documented clearly which helps to understand the model very well. Maintaining the consistency throughout the same function as well as in different functions is the main advantage of this API. This reduces the learning curve and helps the model to switch between tasks like training and inference.

To achieve high performance in our custom dataset the flexibility of adjusting hyper parameters through configuration files or API parameters is very crucial. Without modifying the whole framework yolov8 allows user to adjust his individual components which helps us to experience new techniques and features.

Simple design of yolov8 makes user's works easier and suitable for deployment on edge devices which has only little or limited resources that is very useful for the field of agriculture as well as for the engineers.

6.1.3 Why did we select yolov8?

I. **Real time detection:**

In agriculture environment yolov8 is optimized for the real time practical application. Also it is designed in a way to ensure minimal delay in processing images and detecting diseases.

II. **Speed and accuracy:**

Speed and accuracy of model yolov8 is very high for pre-trained weights and fine-tuned parameters. If we have enough hardware availability to compute the particular task, the speed will increase with the accuracy. Speed and accuracy is high for larger epochs range since, the model sees the data multiple times and learn from it more accurately.

This model offers state of the art accuracy in detecting and classifying objects for the four specific classes (healthy, brown rust, stem rust and loose smut).

III. **Efficiency with custom dataset:**

This yolov8 model is well suitable for working with custom dataset which makes it ideal for our project where we collected around 10,000 images and annotated them accordingly. Augmenting the dataset in this yolov8 environment helped to increase the robustness to our diseases detecting model. Also helps in generalization.

IV. **Ease of integration:**

For the diseases detection system using classification this yolov8 provides us easier integration with smoother development process. Availability of pre trained models helps fine tuning to our custom dataset while pushing existing knowledge to improve performance.

V. **Handling arbitrary sized images:**

It is possible to handle images of any size with newer YOLO architectures, like YOLOv8, without the need for an adaptive pooling layer. Rather, the input image's two dimensions must be multiples of 32 in order for the model to function. This is because the backbone of the network is fully convolutional and has a maximum stride of 32. It is obtained via a sequence of five convolutions, each of which doubles the stride, yielding (2^5 = 32). The desired input size for the photos is defined by the YOLO `imgsz} option, which guarantees that the images meet the required dimensions.

If the dimensions of an image do not naturally fit these requirements, the smaller side is usually padded to match the needed value to the closest multiple of 32. This technique, called "letterboxing," enlarges the image while attempting to keep the aspect ratio as near to the original as feasible. The preservation of the image's spatial relationships and visual characteristics is aided by this technique, which is essential for precise object detection. Therefore, padding is not required and the image can be processed directly by the network as long as the image dimensions are multiples of 32.

Furthermore, square images are favored for model training. This facilitates and expedites the image processing and matrix multiplication processes, which helps to ensure homogeneity throughout the dataset and improve training efficiency. Following these preparation methods enables YOLO to efficiently handle a range of image sizes while preserving aspect ratios and maximizing computing effectiveness.

VI. Model prediction:

Model prediction gives multiple data source compatibility such as individual image, a collection of images or videos. Since we are working with videos as well. Also this increases the performance by processing multiple images or videos in a single batch.

As for the prediction in diverse dimensions in image, Yolov8 trained with image size variable set to 640. Model has an ability to resize the input images dynamically for this fixed dimension. By doing this we can ensure the consistency in the aspect ratio of the image throughout the training process. This operation is done by preserving the aspect ratio of the original image using padding during the resizing process. Padding usually helps in avoiding distortion, ensuring the main features of the image is not lost for more accuracy.

The input images in different sizes are divided into grids by yolov8 and then predictions are made in each grids according to its size. These predictions can adapt to any size of input images. For the diverse dimensions of the image anchor boxes of different aspect ratios is used to detect objects in varying sizes.

VII. **Anchor-free Architecture:**

Anchor-free architecture is one of the main innovations of yolov8 which attracted us to choose yolov8. It eliminates the need for anchor boxes and simplifying the training process. Where Yolov8 predicts the bounding boxes directly and streamlines the training pipeline. This gave less errors in the output.

VIII. **Multi scale predictions:**

one of the main reasons to work with yolov8 was its multiscale predictions that improves accuracy drastically. This multiscale surpasses its predecessors in differentiating small, medium and large objects in an image accurately. Best resulting in scoring diseased images saying how much affected the plants are by diseases. We have used scoring in our model to give the user a useful result saying whether the level of the plant disease is low, medium or high.

6.1.4 How yolov8 differ from other models:

Yolov8 is basically designed for real time object detection or classification. It enhances its speed without sacrificing its accuracy. Computation is more efficient since it has lightweight model structures.

In the training process very simple code and sophisticated data augmentation are used to improve the robustness of the model. Yolov8 shows a high improvement in precision and recall to make it more accurate to classify and detect the objects in the images. Bounding boxes are too very accurate and use refined regression techniques. Overall performance metrics is superior than the other models in terms of speed and accuracy.

6.1.5 Working in yolov8:

A. Training:

Our dataset consists around 10,000 images captured by similar devices in the real world. It was valuable as it represents the actual condition on which our model will operate. Then in the preprocessing step we have augmented the data to increase its diversity and balance the data to fit different models at different times. Commonly used techniques are basic blurring, rotating and flipping, we chose these as they don't mutate the features of the images. Finally, for the classification annotation is done by labeling each images under one of the four classes.

Preprocessing is done to ensure the data is ready to feed into the model for the best performance and accuracy.

We worked with our own custom datasets in the GPU to get more efficiency in the hardware.

Our wheat dataset has nearly 10,000 images for four classes such as healthy, brown rust, stem rust and loose smut. We trained our yolov8 model for this dataset in 100 epochs. Changed the runtime mode to GPU and worked in it. We divided the data into ratio 70:20:10 and fed 70%of the data to train the model. Key training setting done in learning rate, weight decay and batch size. Additionally, the choice of the optimizer and loss function can impact the training process. Augmentation is done to train the model better for unseen data as well.

B. Validation:

precision is used to get accurate metrics like mAP50, mAP75 and mAP95 to evaluate the model. 20% of the dataset is used to validate the model. Here model remembers their training configurations for straightforward validation. We didn't pass
any argument specifically as the model retains its training data and arguments as model attributes such as input image size and performance thresholds.

6.2 Model Inceptionresnetv2

InceptionResNetV2 is a state-of-the-art deep convolutional neural network architecture, combining the architecture of Inception with the strength of residual connections, making it an effective approach to image classification problem. In addition, there are models that are well-known to work with multi-scale features such as Inception architecture. This design enables to deal with a broad range of levels of detail within the same layer, so that the network can learn feature representations that are more rich and diverse.

Residual connections introduce shortcut connections which skip one or more layers thus, allowing the gradient to propagate faster through the layers during backpropagation. that can be trained to much greater depths leading to less risk of overfitting and better generalization on unseen data.

InceptionResNetV2 combines the advantages of Inception and ResNet, which makes it deep and wide and allows it to learn complex and hierarchical representations of input data. Hence, it is very useful in complex classification tasks.

6.2.1 Inceptionresnetv2 architecture:

InceptionResNetV2 is a technique for an extravagant neural network architecture that combines the advantages of the Inception modules with straight connections via residual learning. This architecture is developed in order to tackle the challenges of deep learning including that of training very deep networks and capturing complex features at multiple scales for achieving high performance and robust image classifications.

InceptionResNetV2 is an extended version of previous inception network designed for more deep networks. In Inception Models, an Inception module is a collection of convolutional filters of different sizes (1x1, 3x3, 5x5) and then fed in parallel to the input. The results from these filters are concatenated to create an input for the next layer. This design we come up with end up with way, the network is able to get features from multiple scale even in the same layer which gives us very rich and diverse representation for the data its see.

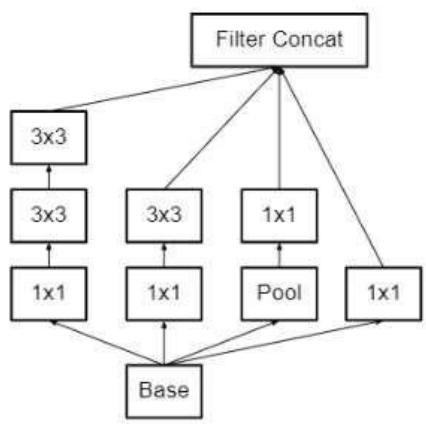


Figure 6. 4 Convolution filters

The architecture consists of a stem for initial feature extraction. followed by global average pooling, and a fully connected layer for classification. This configuration makes InceptionResNetV2 capable of effectively addressing large and challenging data tasks for classification tasks.

6.2.2 Ease of interface:

InceptionResNetV2 offers a user-friendly interface that makes it accessible and practical for agricultural applications such as wheat disease detection. Its robust architecture allows for efficient training and high accuracy in distinguishing between healthy and diseased wheat plants, and among the specific diseases of brown rust, stem rust, and loose smut. This ease of use is crucial for implementing the model in real-world scenarios to help farmers.

The model can be trained using personal device-captured 10,000 images while the pretrained weights on large datasets would have them fine-tuned by excluding top classification layers adding custom classification layers, which makes the training process efficient, swift deployment and seamless.

Training the model on our custom dataset can be achieved by using a simple ‘fit’ function and evaluated. Then, it can be used for inference.

In addition, its support to popular deep learning frameworks including TensorFlow and Keras makes it easy for integration with contemporary agricultural monitoring networks, so farmers can use it in combination to optimize their treatment plans and contribute to financial sustainability.

6.2.3 Why did we select Inceptionresnetv2:

I. **Strong performance:**

Many high-level attributes can be successfully extracted by InceptionResNetV2 during recognition. It is ideal for classifying different types of wheat diseases because of its ability in capturing multiscale features by inception modules and enabling efficient training of deep networks by use residual connections.

II. **Accuracy:**

For image classification tasks this model is specialized in high accuracy. Inceptionresnetv2 has an ability to extract multi scale and complexity features. It can effectively distinguish between healthy and diseased wheat with further classifications of diseases with high accuracy in overall 100 epochs.

III. **Transfer learning:**

With InceptionResNetV2 already trained on large datasets such as ImageNet, transfer learning can be done faster. Fine-tune the pre-trained model using your wheat image data set, and take advantage of learnt features to reduce training time and amount of data needed for good performance.

IV. **Architecture design:**

To balance the computation efficiency and model complexity the architecture of Inceptionresnetv2 has been designed carefully. Most importantly projects like ours, for the real time application this model is very useful.

V. **State of art performance:**

For image classification one of the best model representations was Inceptionresnetv2.

For comparison, classification this model provides a strong base for ant task. Such as wheat detection tasks like ours.

VI. **Efficiency:**

Provides a good balance between model complexity and computational efficiency. This is crucial for applications such as image classification that makes it adaptable for real world application. The model architecture, with native modules and residual interfaces, allows efficient separation of functions and keeps computational requirements manageable.

Its ability to process images quickly and provide accurate results could be valuable to farmers looking to diagnose and treat diseases in wheat crops, ultimately improving yields and financial stability.

6.2.4 Working in Inceptionresnetv2:

Our project was objected to detect diseases in the wheat plant using classifications. We used this Inceptionreznetv2 model which eases our work with high performance and accuracy.

A. **Preprocessing:**

First step we used in preprocessing was to resize the JPEG images into approximately 640*840 pixels. Maintain a uniform sizes for the images help to maintain the constancy throughout the process, which is very important in training process. Resizing will overwrite the original image. This preprocessing steps ensures the accuracy in training.

To generate the dataset Tensorflow API is used. Here we have done data augmentation to increase the diverse of the image dataset for better robustness. This generator also scales the images in the range of [0,1] to normalize the dataset to improve speed and the performance.

Although the default input size of the Inceptionrenetv2 model is 299*299 we have used a high pixel dimension of 640*840 to get better extraction of the diseased images. Because most of the images of the wheat crops are taken from various distances or angles or resolutions and using high resolution pixels would help the model in learning the image that will be useful for our project to get a high accuracy.

B. **Model preparation:**

Inceptionresnetv2 contains pre-trained weights which helps in best feature extraction. In our code we have excluded top classification layers for our custom dataset and added custom layers on behalf to specifically classify our diseases.

Secondly, weights of each layers are not upgraded in during training, which is very useful to memorize the learned features without any loss and focusing on the custom classification layers. Custom layers used are, pooling layer, dense layer, ReLU activation and to prevent overfitting additionally a dropout layer and most importantly, for multi class classification a softmax activation.

C. **Training:**

As usual from the dataset containing around 10,000 images, a split of 70% is fed to the training. And for the multi class classification a special loss function ‘categorical cross entropy’ is used. Function ‘fit’ is used to evaluate the training and validation process followed by running 100 epochs to get best performance and accuracy.

6.3 Model Resnet50

Resnet50 is an image classification model that belongs to ResNet family introduced in 2015. Since then many new classification model have been introduced because of the strong performance that Resnet50 achieved. Default ResNet models can identify any of 1000 classes in the ImageNet dataset.

Degradation was the primary problem that ResNet solved where, when the network goes deeper their accuracy saturates easily. This degradation is not because of the overfitting but difficulty of optimizing the training process.

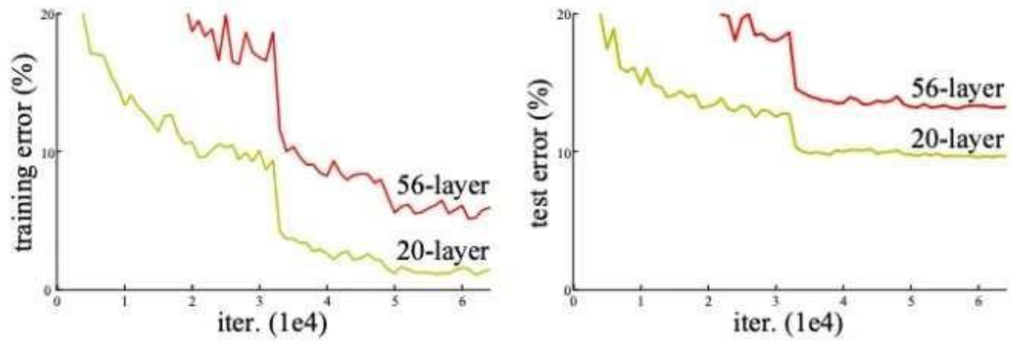


Figure 6. 5 Performance of Resnet

Usually all of the ResNet models solves a problem using residual blocks that skips the connection in between the flow of the information. ReLU function activated to introduce non linearity in to the data that deals with custom dataset better. Then comes the bottleneck convolution layer with 3 layers with the filter sizes of 1*1, 3*3, 1*1 accordingly to help with normalization.

Throughout the integration of bottleneck layers and skip connection Resnet50 reduces the gradient issue, providing a best profound model for image classification.

6.3.1 Resnet50 Architecture:

ResNet-50 is made up of 50 layers split up into 5 blocks, each of which has a collection of residual blocks. The network can learn more accurate representations of the input data by using the residual blocks, which enable the preservation of information from previous levels.

a. **Convolution layers:**

This is the first layer which performs convolution to the input image. Then comes the max pool layer that down samples the output of the first layer and then sent to the residual blocks

b. **Residual blocks:**

Consists of two convolution layers, each of them have a normalization layer and ReLU function that comes after. Output of the Residual blocks passed to upcoming blocks.

c. **Fully connected layer:**

Final layer of this network. Takes the output from the residual blocks and pass it to the final output classes. For the number of the classes that we have in our task is equal to the number of neurons we use in this layer.

Another key feature of this Resnet50 architecture is the skipping connection. This connection saves the details of the previous layers and helps the model to learn better about the input data.

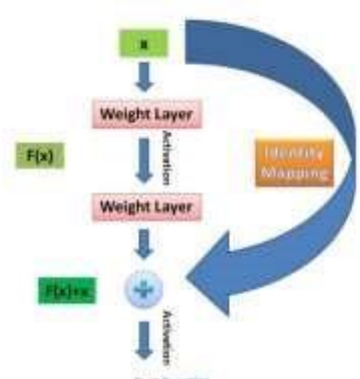


Figure 6. 6 Fully connected layer

6.3.2 How Resnet50 differ from other models:

One of the main advantages is, it has the ability to train very deep networks that have more than 100 layers.

This model has skip connection that helps to preserve the details of the previous layers.

This also achieves the state of the art results in the image classification tasks widely.

6.3.3 Ease of interface:

Provides a user friendly interface with feasible deep learning frameworks like PyTorch and Tensorflow for quick set up. A pre trained model that helps to fine tune our custom dataset using learned features for high performance.

For the loading, training and modification both Tensorflow and PyTorch provide a straight API.

Training, validation and inference are well documented. This provides high efficiency. Consistency across all the functionalities provided by this API which allows user to focus on classifying wheat diseases.

All the codes, explanation of parameters and practices are available in the Resnet50 documentation which gives you a better understanding using that.

Using matplotlib or tensorboard can understand the performance better. Additionally, for our custom dataset adjusting hyper parameters, and other settings through API parameters gives a flexibility for fine tuning the model.

6.3.4 Why did we select Resnet50:

I. **Performance**

Resnet50 itself is a model that designed to perform best for various classification tasks with high accuracy. This architecture provides us a reliable approach towards our project.

II. **Feature extraction**

Starting from the first layer till the deeper layers, each layer captures different type of features from the image either a simple one or complex features. This ability of extracting features in a hierarchical manner gives the better accuracy in classifying diseases of the wheat.

III. **Pre-trained model**

Pre-trained weights helped us to reduce the training time that wanted to achieve high accuracy. Also transfer learning increased the model's performance when working with our 10, 000 images.

IV. **Architecture**

By extracting complex features of the images, deep layers of Resnet50 helped to classify between healthy and diseased images as well as between various disease images. Deep layers of the Ressnet50 provides high accuracy in training. Residual connections in other hand prevents gradients from vanishing to ensure stability of the model by training the deep networks.

6.3.5 Working on Resnet50:

Our dataset contains 4 classes of around 10,000 images captured by our own devices. Used splitting in a ratio of 70:20:10 for training, validation and testing accordingly. The chosen image size was approximately 640*840 for better resolution. Images are labeled using categorical encoding to handle multi class classification.

A. **Data preprocessing:**

After loading the data, we have enhanced the data using augmentation. Followed by freezing the pre-trained layers to allow the model to learn the custom layers without altering the existing features. custom layers used here are flatten layer to convert 2D image into 1D and dense layers to classify the images.

categorical cross entropy was selected here as a loss function for our multi class classification to measure the output performance between 0 and 1.

B. **Training:**

Training data generator used to train the model while validation data generator monitors the performance of the unseen data.

Training involved two iterations, one for epochs and the other one by batch. The image is fed as an input to calculate loss function and by using an optimizer, updating the gradient of the training parameters. In our project epochs were set for 100.

To simplify this training process, high level model API is used. Train API parameters includes epochs and batch in it.

File of the trained model is used to predict the class of the new image by just calling ‘eval API’ of the model.

6.4 Challenges faced:

Hand labeling for the 10,000 images for our custom dataset was the biggest challenge for us while training our model for its accuracy and consistent. Where 5,000 of these images were extracted from videos to create more real world images.

For the imbalanced classes where the model becomes biased for most frequent classes we used augmentation. To augment the data using effective techniques that won't include any artifacts which could mislead the training process was another crucial process.

Understanding the model architecture and it is parameters, to make the model fit our custom dataset, we have trained tens of models which we just learned them our mistakes, after iterations of data preprocessing and understanding the working of the model, we felt comfortable with our final findings.

Since our model was one of the large models in computer visions it was computationally intensive. It required powerful GPUs and slowed down the training process most of the time. We could purchase an additional GPU for monthly based do our work easily than before to overcome this

Choosing the appropriate evaluation metrics was a challenge while receiving the output. We started with plotting the training accuracy and then train/Val loss, later by plotting confusion matrix for the accuracies, we got a better understanding about the model. It was an easy to find the mistakes that we have done. Other metrics were helpful too such as precision, recall.

Detecting complex images such as varying scales or different aspect than the usual ones were some more challenges we faced during the training. Especially when using a pre-trained model like yolov8 to a custom dataset requires a careful handling. Starting from labeling till training, to avoid catastrophic forgetting while grasping the learned features.

For Inceptionresnetv2, the default input size was 299*299 and since we used 640*840 which was high resolution, the model started to capture the noises and other irrelevant features as well. Adding more dropout layers for regularization and data augmentation helped us to overcome this challenge.

Chapter 7 : Results

There were 10% of each model's dataset randomly selected from the whole dataset which the model never saw it neither in training nor in validation process. These images were used to test each model, besides that other datasets from Kaggle were used to test the model.

These three models which were the main models (healthy vs diseased, rust vs smut and brown rust vs stem rust) showed high accuracy nearing perfection when tested using the subset of the dataset which the team collected.

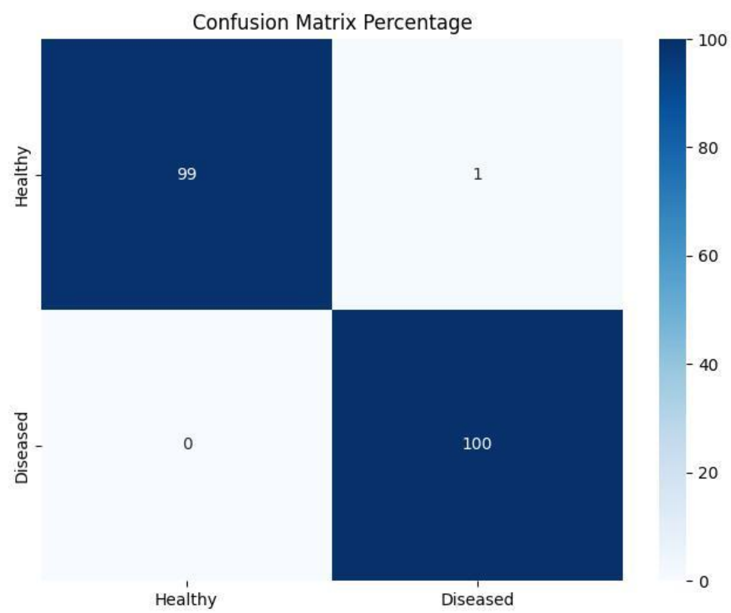


Figure 7. 1 Healthy vs diseased model tested on 10% of the collected data

The context of this high accuracy lies in the dataset itself and the preprocessing, the dataset has been collected by the model trainers which had the view of making regularization from the beginning, while making sure the normal constrains and changes of the nature are represented in the dataset, on the other hand the disease images were made sure that trained human eyes can identify them.

Labeling and preprocessing were kept through the entire journey, while multiple approaches of preprocessing the dataset were discarded after they showed not just poor performance but also poor representation of the reality which the model will be tested. Models and their parameters were selected after learning from several models which were trained and tested, so that the final selection model ends up to be one which can process the image as the reality of images is while also utilizing the capabilities of computer vision.

While there are limited wheat image datasets available on the internet, the datasets which exemplify the reality scenario of taking wheat images are even lesser. The model has been tested with some of the Kaggle dataset which we will comment on the model predictions.

Wheat leaf is image dataset of wheat leaves from Holeta wheat farm in Ethiopia, captured using a high-resolution Canon EOS 5D Mark III camera. It consists of 1,266 images categorized into healthy, Stripe Rust infected, and Septoria infected leaves, with 102 healthy, 208 Stripe Rust, and 97 Septoria cases explicitly mentioned. Models which were trained in this dataset such as inceptionV3 has given 90% accuracy.

The first model showed the below mentioned results when tested with this dataset. While the model shows poor performance the investigations done into the wrongly identified images shows a certain anomaly.

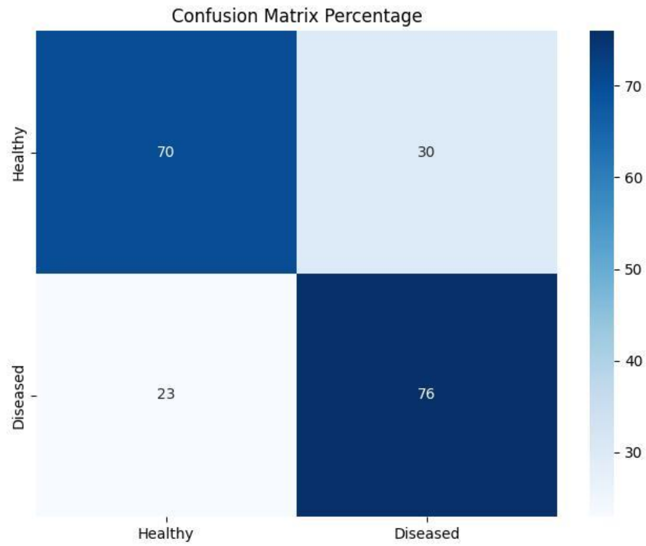


Figure 7. 2 Healthy vs diseased model tested on Wheat leaf dataset

The images of this dataset consists of three classes which 2 of them didn’t include the classes which the model was trained, regardless of that the test are done due to some similarities between the two disease classes which are in this dataset and the disease classes which our model has trained.

Another important point for clarification is that the background of these image datasets are not regularized, many of the images were taken in real scenario of wheat plant growing in the field, while others were leaves which were plucked and photographed on another objects background such as clothes, the model which we had mostly failed to identify these, as we worked exclusively on images which were taken in field.

Figure 7. 3 Wrongly predicted images due to background

While the most of the images which the model predicted wrong were such kind of images, there were few images from the test dataset which we had and the Kaggle datasets which had shown issue of sensitivity which the model is suffering from. The image dataset which had been collected by our team is from fields which had stable climate conditions during the times of visit which cause a phenomenon that the leaves didn’t had any climate issues which causes them to develop marks which might appear diseases.

Figure 7. 4 Wrong detection which shows model sensitivity

The second main model which has been trained was to classify between rust and smut, two main categories of wheat diseases, the results of the model when tested with the two datasets is mentioned below.

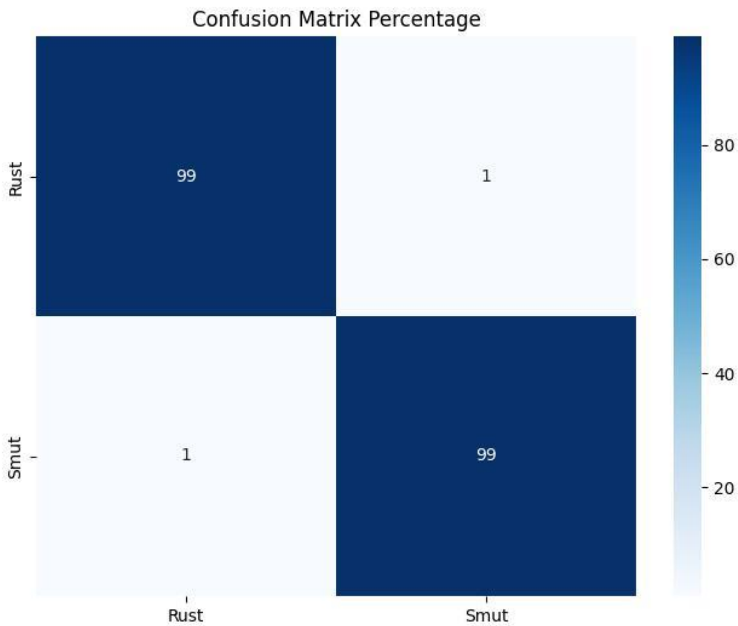


Figure 7. 5 Rust vs Smut model tested with the data that we collected

When the same model was tested using the wheat leaf dataset it showed similar promising results which shows that the model is working fine.

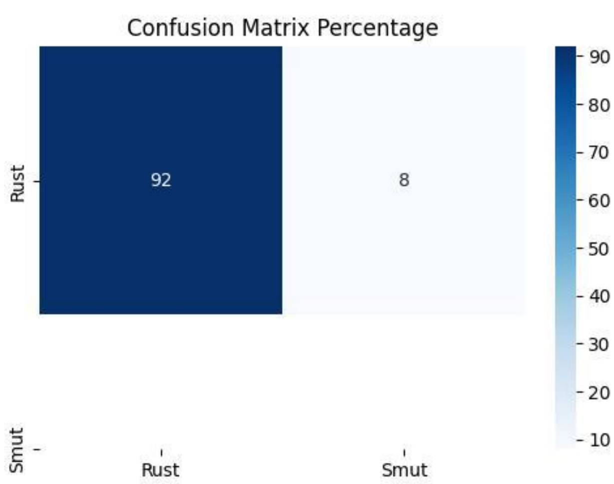


Figure 7. 6 Rust vs Smut model tested with wheat leaf dataset

Considering that most of the wrongly identified images has background issues such as the ones mentioned before, stripe rust was not trained in our model but it is pretty much similar to brown rust which is the main leaf rust which the model has learned, in fact its highly difficult for even farmers to classify between these two in naked eye, this shows that the model is able to generalize like a human eye, by detecting stripe rust which wasn't even trained in it as brown rust.

Third model which had been trained is to classify between brown rust and stem rust, there was no other valuable dataset to test this model so we rely on our own dataset, which shows an ideal classification results.

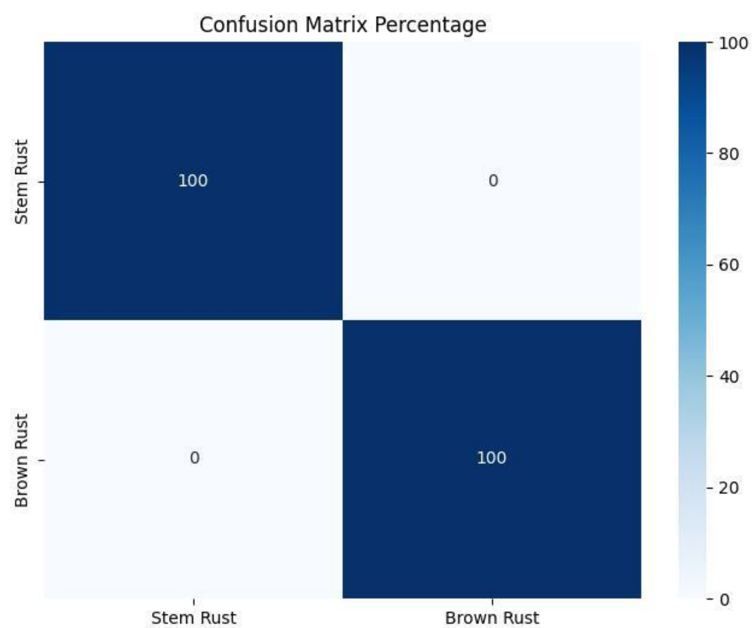


Figure 7. 7 Brown rust vs Stem rust tested with the data that we collected

This accuracy is believable due to the nature difference of these two diseases, while brown rust and stem rust are both in the family of rust but the first one effects the leaf while the second effects the stem, they spread in other parts of the plant when the disease in its last stages. This difference makes them easy to be identified from each other.

We covered the testing results of three models so far which are the main one.
There are three more models which we trained and tested, these models are subjective models which classify each of the three diseases which we had in the dataset which we collected into stages, high medium and low.

We got enough support from the agriculture scientists in both Tando jam Agriculture University and Pakistan agriculture research and crop disease (PARC) and the supporting materials which they provided, the images of each disease were labeled and trained accordingly.

Labeling the images for scoring models was highly subjective, and the testing results of the model shows similar.

The skewness of the available data also has a big role on this, each diseased class has been distributed in to high, medium and low, according to the available data, that notion might be different if more images of that disease were available.

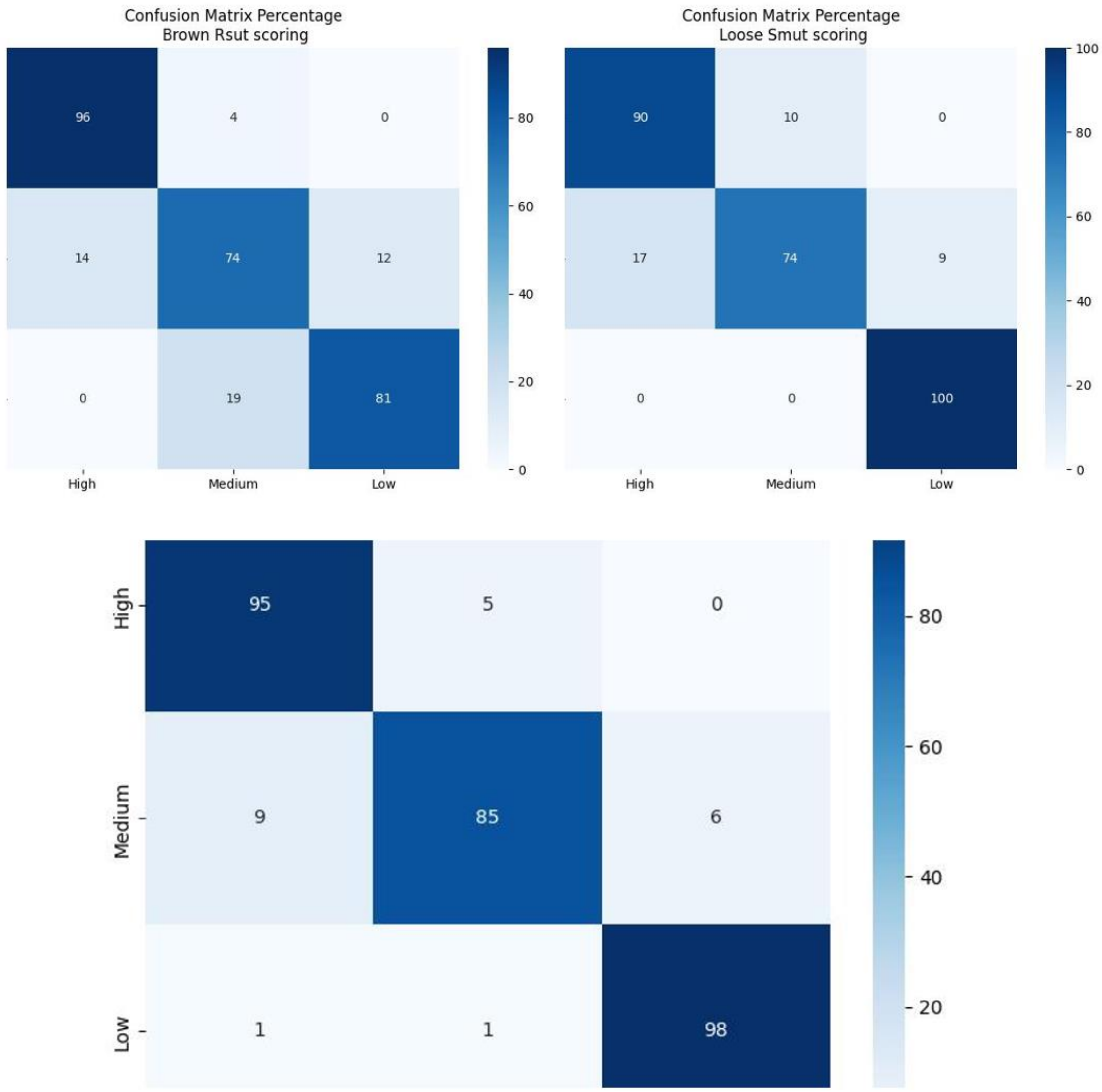


Figure 7. 8 3 Scoring models tested with the data that we collected

Table 7. 1 Overall Model Validation Measurement

Model		Accuracy	precision	Recall
Healthy vs. Diseased		99.9	99.8	100
Rust vs. Smut		99.9	99.8	100
Brown Rust vs. Stem Rust		100	100	100
Stem Rust Scoring	High	95.7	90.5	91.3
	Medium	95.7	95.8	91.9
	Low	95.7	90.8	97.1
Brown Rust Scoring	High	85.3	90	94.7
	Medium	85.3	77.3	77.3
	Low	85.3	87.6	82.5
Loose Smut Scoring	High	94.6	95.2	93
	Medium	94.6	91.6	93.6
	Low	94.6	97.6	97.6

Chapter 8 : Integration and Interface

8.1 Model integration:

Integrating YOLO v8 from Ultralytics into a Python environment is a straightforward process that enhances the capabilities of any computer vision project, such as those built for detecting wheat diseases. The integration begins with installing the YOLO v8 package via pip, the Python package manager. This can be accomplished by running the command in the terminal. Once installed, the package can be imported into the Python code. This allows for utilizing YOLO v8's model capabilities within the Python environment, enabling high-precision detection and classification tasks.

YOLO v8's compatibility with Django, a robust web framework for Python, ensures seamless integration into web-based applications. Django's architecture supports the inclusion of complex backend systems and sophisticated frontend interfaces. Incorporating YOLO v8 into a Django project can leverage the framework's scalability and ease of integration. This enables the creation of dynamic web applications capable of processing and analyzing images in real time, providing a rich user experience through interactive and responsive interfaces. The synergy between YOLO v8's advanced computer vision functionalities and Django's comprehensive web development features makes it an ideal combination for developing innovative and efficient web applications.

8.2 Framework Choice and Justification:

For this project, the model selected Django as the primary framework due to its scalability and widespread adoption in the industry. The system benefits from Django's robust architecture and comprehensive documentation, making it an ideal choice for developing web applications that require seamless integration with complex backend systems. Its compatibility with Python facilitated smooth integration with the computer vision model, enabling efficient handling and processing of data.

8.3 Frontend Development:

To create an intuitive and responsive user interface, the system utilized a combination of HTML, CSS, and JavaScript. HTML provided the foundational structure of the web pages, while CSS enhanced visual aesthetics, ensuring a clean and modern design. JavaScript implemented dynamic functionalities, enabling interactive elements and real-time updates on the frontend.

The frontend was designed with a focus on usability and clarity, ensuring that users can easily navigate through the application and access essential features. Users can upload images, video or directory of wheat plants, which the model processes to detect potential diseases. The results, including identified diseases and their severity stage, are presented in a user-friendly manner.

By leveraging Django's powerful backend capabilities and combining them with a carefully crafted frontend, the system offers a comprehensive solution for wheat disease detection. This integration not only enhances the performance and reliability of the application but also ensures a seamless and engaging user experience.

8.4 Uploading Images, Videos, and Directories:

Users can upload images in various formats and resolutions, also videos of different formats are supported, users can also upload directory paths with multiple images to get inferences related to all of them.

The code preprocesses all of these media files assuring each one of them is processed in a way that reflects it, at the same time a way that gets the best inference possible out of the technology.

The dimensions of each image whether it is coming into the system as a single image or bundle of images get rounded to the nearest multiple of 32, which will down-sample or up sample height and width using the openCV library, the new dimensions are then passed to the inference model with the image source assuring that the model sees the image as it is without changing the aspect ratio or resizing the image to fit predefined dimensions. This practice gave the most consistent and realistic prediction throughout the testing.

The videos are read using the openCV library, and the total number of frames in the video is calculated and shown to the user with the output to let the user know how many total images in the video were processed. As video processing is more time-consuming the user has to select which specific inference model among the models that we have they want to select. The video and the selected model are passed to the backend for inference which will return the corresponding predictions.

The directory path can be influenced by the model, allowing the user to paste the path and select the model which they want to run the inference, only the images in the directory will be applicable for inference and the output is going to be a count of number of images in each label of the selected model.

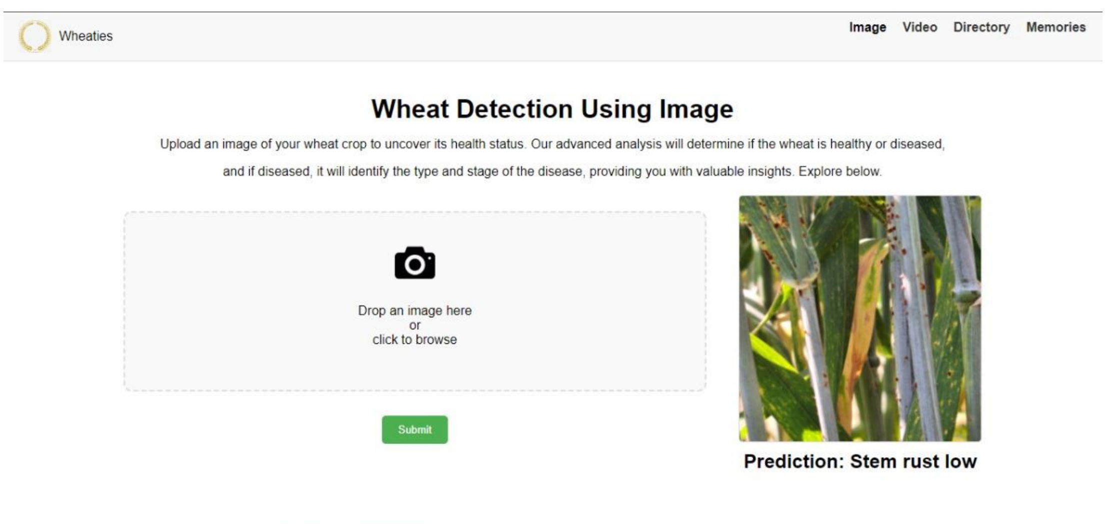


Figure 8. 1 User uploaded image

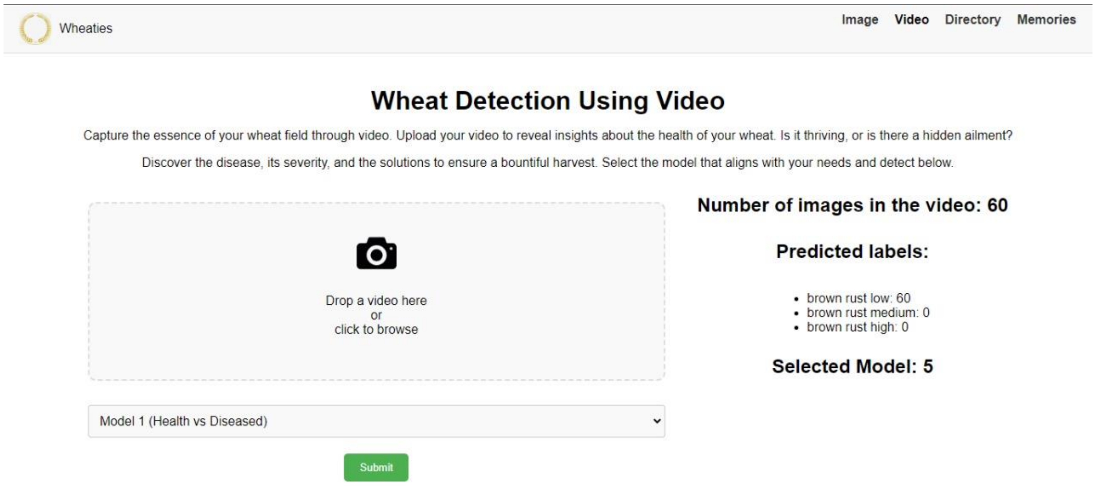


Figure 8. 2 User uploaded video

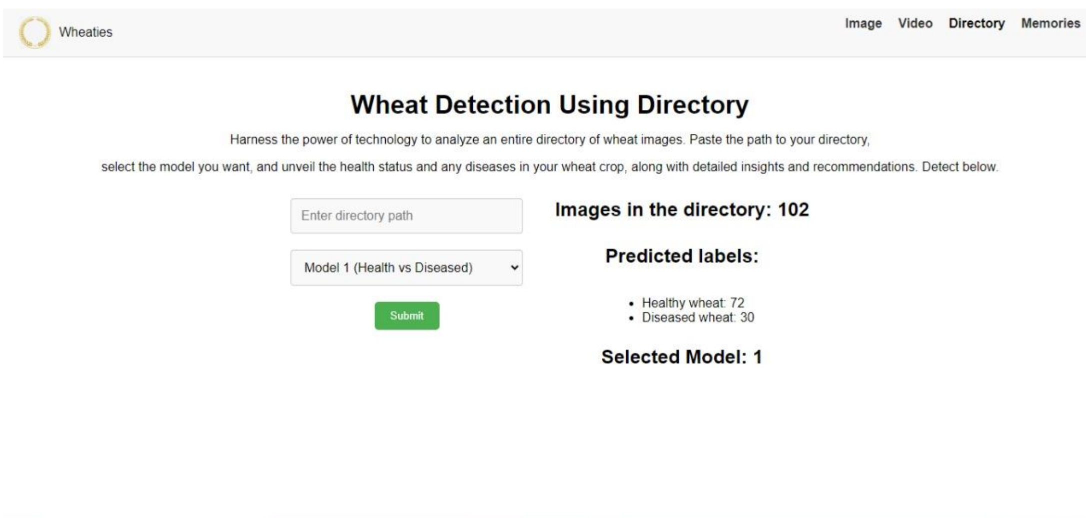


Figure 8. 3 User uploaded directory

8.5 Reporting Results:

Single images are passed to all the models and will return inference at the end which will either be healthy or in terms of disease the type and the stage of that disease. The input image will also be shown in the interface.

For videos and directories user has to select the desired model which they want to use and it will return the count of the number of images or frames in each of the labels of the model that has been selected,

If the user wants to run inference of more than one model for a video or directory they have to do each model once at a time.

Conclusion

Summary

As we talk about the wheat farming in rural Pakistan our project, Deep learning based Rust and Smut disease detection paving a high way for the economy growth as well as to reduce the poverty by increasing the crop yields. By using some useful as well as important AI tools and Deep learning techniques this detector will give an early detection of the diseases which saves time from harvesting diseased crops and from economy crisis. This helps managing Rust and Smut diseases and prevents major wheat crop threats.

Our model has ability to classify the diseases and to detect these diseases in its 3 stages (low, medium and high) accurately. This helps farmers to take actions at the correct time by preventing the crops from dying forever. Also the user-interface that can be accessed by any farmers easily and learn about the diseases that are described and prescribed by the agriculture expertise gives more advantages. Which leads to an economic stability in all over Pakistan.

Our project improves food security a big milestone, by increasing the food production and decreasing the economy loss. Moreover, an AI bases much accurate solution that farmers could access too gives a new side for both the technology and agriculture field as a combination.

The success of this project is based on combining the latest technology with the traditional farming lifestyle which overcomes the challenges in both agriculture and technology sectors. Leading a better future to Pakistan.

Future Improvements:

The constrains on this project were using a dataset which the team has collected by themselves, this brings up more constrains such as wheat being annually grown crop, limitations of area which was possible to be covered and not being able to visit fields more often.

This projects can be improved overtime, if different prospective of taking pictures were considered, while regularization of the dataset is crucial, the dataset can be enlarged while still maintaining regularization which will cause the models to generalize more.

Robust dataset will make robust models, a big challenge for the team was to find representation of diseased images in the dataset, and a bigger challenge is that wheat diseases vary according to climate of the region, collecting a dataset which represent wider area will enhance the model.

Further collaboration from the agriculture teams specially during the scoring labeling will improve the robustness of the model as well as the credibility.

While working with agriculture scientists we saw the gap between them and the technology, they value guidance which comes from pure statistical calculations way more than what a model can do. This might seem a hurdle but it can be a huge advantage if the two fields were bridged correctly, agriculture scientists will never appreciate the complexity of models, technology background people shouldn’t preach to teach them the significance, rather we suggest the outcomes of these models should be combined with the knowledge of agriculture to make predictions which guides the farms into actions.

Farmers and agriculture scientists want does’ and don’ts; they want direct guidance from technology.

If we add access to more information such as weather, location, field related information, wheat variety and the specific area of the field which a disease has been spotted using these models, then deducting guidance inform of actions will be easy, agriculture scientist and farmers will rely on these technology, and technology can get more robust while both fields are working together.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[2] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[3] W. Fang, P. E. D. Love, H. Luo, and L. Ding, “Computer vision for behaviour-based safety in construction: A review and future directions,” *Advanced Engineering Informatics*, vol. 43, p. 100980, Jan. 2020, doi: 10.1016/j.aei.2019.100980.

[4] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2016.

[6] M. Elgendy, *Deep learning for vision systems*. Shelter Island, NY: Manning Publications Co, 2020.

[7] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Computer Vision – ECCV 2014*, vol. 8689, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8689. , Cham: Springer International Publishing, 2014, pp. 818–833. doi: 10.1007/978-3-319-10590-1_53.

[8] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, “Review of Image Classification Algorithms Based on Convolutional Neural Networks,” *Remote Sensing*, vol. 13, no. 22, p. 4712, Nov. 2021, doi: 10.3390/rs13224712.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[10] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 10, 2015. Accessed: Feb. 24, 2024. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[12] F. Sultana, A. Sufian, and P. Dutta, “A Review of Object Detection Models Based on Convolutional Neural Network,” in *Intelligent Computing: Image Processing Based Applications*, vol. 1157, J. K. Mandal and S. Banerjee, Eds., in Advances in Intelligent Systems and Computing, vol. 1157. , Singapore: Springer Singapore, 2020, pp. 1–16. doi: 10.1007/978-981-15-4288-6_1.

[13] A. Dhillon and G. K. Verma, “Convolutional neural network: a review of models, methodologies and applications to object detection,” *Prog Artif Intell*, vol. 9, no. 2, pp. 85– 112, Jun. 2020, doi: 10.1007/s13748-019-00203-0.

[14] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks.” arXiv, Dec. 02, 2015. Accessed: Feb. 24, 2024. [Online]. Available: http://arxiv.org/abs/1511.08458 [15] M. H. Hossen, M. Mohibullah, C. S. Muzammel, T. Ahmed, S. Acharjee, and M. B. Panna, “Wheat Diseases Detection and Classification using Convolutional Neural Network (CNN),” *IJACSA*, vol. 13, no. 11, 2022, doi: 10.14569/IJACSA.2022.0131183.

[16] R. Rajesh Kanna and V. Ulagamuthalvi, “Identification of wheat plant disease using hybrid model of pre-processing with CNN,” *IFS*, pp. 1–11, Oct. 2023, doi: 10.3233/JIFS233672.

[17] V. Kukreja and D. Kumar, “Automatic Classification of Wheat Rust Diseases Using Deep Convolutional Neural Networks,” in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India: IEEE, Sep. 2021, pp. 1–6. doi: 10.1109/ICRITO51393.2021.9596133.

[18] A. Singh and M. Arora, “CNN Based Detection of Healthy and Unhealthy Wheat Crop,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India: IEEE, Sep. 2020, pp. 121–125. doi: 10.1109/ICOSEC49089.2020.9215340.

[19] A. Hussain, M. Ahmad, I. A. Mughal, and H. Ali, “Automatic Disease Detection in

Wheat Crop using Convolution Neural Network,” 2018, doi: 10.13140/RG.2.2.14191.46244.

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

[21] L. Xu *et al.*, “Wheat leaf disease identification based on deep learning algorithms,” *Physiological and Molecular Plant Pathology*, vol. 123, p. 101940, Jan. 2023, doi:

10.1016/j.pmpp.2022.101940.

[22] M. Dong *et al.*, “Novel method for identifying wheat leaf disease images based on differential amplification convolutional neural network,” *International Journal of Agricultural and Biological Engineering*, vol. 13, no. 4, pp. 205–210, 2020, doi:

10.25165/j.ijabe.20201304.4826.

[23] M. D. Bolton, J. A. Kolmer, and D. F. Garvin, “Wheat leaf rust caused by Puccinia triticina,” *Molecular Plant Pathology*, vol. 9, no. 5, Art. no. 5, 2008, doi: 10.1111/j.13643703.2008.00487.x.

[24] A. P. Roelfs, R. P. Singh, and E. E. Saari, *Rust Diseases of Wheat: Concepts and Methods of Disease Management*. CIMMYT, 1992.

[25] R. P. Singh *et al.*, “Will Stem Rust Destroy the World’s Wheat Crop?,” in *Advances in Agronomy*, vol. 98, Academic Press, 2008, pp. 271–309. doi: 10.1016/S00652113(08)00205-8.
[26] I. U. Din *et al.*, “Genetic Characterization of Advance Bread Wheat Lines for Yield and Stripe Rust Resistance,” *ACS Omega*, vol. 8, no. 29, Art. no. 29, Jul. 2023, doi:
10.1021/acsomega.3c01981.
[27] “IDENTIFICATION OF ISOLATED STRAINS OF USTILAGO TRITICI RS-01 FROM AN INFECTED WHEAT CROP | Badar | Pakistan Journal of Phytopathology.” Accessed:
Feb. 24, 2024. [Online]. Available: https://pakps.com/pjp/index.php/PJP/article/view/1038
[28] L. H. Purdy, “Flag smut of wheat,” *Bot. Rev*, vol. 31, no. 4, Art. no. 4, Oct. 1965, doi:
10.1007/BF02858609.
[29] M. McMullen *et al.*, “A Unified Effort to Fight an Enemy of Wheat and Barley: Fusarium Head Blight,” *Plant Disease*, vol. 96, no. 12, Art. no. 12, Dec. 2012, doi: 10.1094/PDIS-0312-0291-FE.
[30] M. R. Khan, M. Imtiaz, F. Farhatullah, S. Ahmad, and S. Ali, “Northern Himalayan Region of Pakistan with Cold and Wet Climate Favors a High Prevalence of Wheat Powdery Mildew,” *SJA*, vol. 35, no. 1, 2019, doi: 10.17582/journal.sja/2019/35.1.187.193.
[31] P. A. Nazarov, D. N. Baleev, M. I. Ivanova, L. M. Sokolova, and M. V. Karakozova,
“Infectious Plant Diseases: Etiology, Current Status, Problems and Prospects in Plant Protection,” *Acta Naturae*, vol. 12, no. 3, pp. 46–59, 2020, doi: 10.32607/actanaturae.11026.
[32] A. Gamliel, “High Consequence Plant Pathogens,” in *Crop Biosecurity*, M. L. Gullino, J. Fletcher, A. Gamliel, and J. P. Stack, Eds., in NATO Science for Peace and Security Series C: Environmental Security. Dordrecht: Springer Netherlands, 2008, pp. 25–36. doi:
10.1007/978-1-4020-8476-8_4.
[33] M. Vurro, B. Bonciani, and G. Vannacci, “Emerging infectious diseases of crop plants in developing countries: impact on agriculture and socio-economic consequences,” *Food Sec.*, vol. 2, no. 2, pp. 113–132, Jun. 2010, doi: 10.1007/s12571-010-0062-7.
[34] J. K. M. Brown and M. S. Hovmøller, “Aerial Dispersal of Pathogens on the Global and Continental Scales and Its Impact on Plant Disease,” *Science*, vol. 297, no. 5581, pp. 537– 541, Jul. 2002, doi: 10.1126/science.1072678.
[35] M. L. Bhardwaj, Rohitashw Kumar, and H. R. Gautam, “Climate change and its impact on plant diseases,” *current science*, vol. 105, no. 12, pp. 1685–1691, Dec. 2013.
[36] S. Savary *et al.*, “International Agricultural Research Tackling the Effects of Global and Climate Changes on Plant Diseases in the Developing World,” *Plant*

Disease, vol. 95, no. 10, pp. 1204–1216, Oct. 2011, doi: 10.1094/PDIS-04-11-0316.

[37] P. P. Anderson, “The Future World Food Situation and the Role of Plant Diseases,” *PHI*, 2001, doi: 10.1094/PHI-I-2001-0425-01.

[38] J. Zhang *et al.*, “Monitoring plant diseases and pests through remote sensing technology:
A review,” *Computers and Electronics in Agriculture*, vol. 165, p. 104943, Oct. 2019, doi:
10.1016/j.compag.2019.104943.

[39] A. S. Moffat, “Finding New Ways to Fight Plant Diseases,” *Science*, vol. 292, no. 5525, pp. 2270–2273, Jun. 2001, doi: 10.1126/science.292.5525.2270.

[40] M. Figueroa, K. E. Hammond-Kosack, and P. S. Solomon, “A review of wheat diseases—a field perspective,” *Molecular Plant Pathology*, vol. 19, no. 6, pp. 1523–1536, Jun. 2018, doi: 10.1111/mpp.12618.

[41] M. S. Randhawa, S. Bhavani, P. K. Singh, J. Huerta-Espino, and R. P. Singh, “Disease
Resistance in Wheat: Present Status and Future Prospects,” in *Disease Resistance in Crop Plants: Molecular, Genetic and Genomic Perspectives*, S. H. Wani, Ed., Cham: Springer International Publishing, 2019, pp. 61–81. doi: 10.1007/978-3-030-20728-1_4.

[42] S. Bhavani *et al.*, “Globally Important Wheat Diseases: Status, Challenges, Breeding and
Genomic Tools to Enhance Resistance Durability,” in *Genomic Designing for Biotic Stress Resistant Cereal Crops*, C. Kole, Ed., Cham: Springer International Publishing, 2021, pp. 59–128. doi: 10.1007/978-3-030-75879-0_2.

[43] Julius Kühn-Institute (JKI), Institute for Resistance Research and Stress Tolerance,
Germany *et al.*, “Wheat diseases: an overview,” in *Burleigh Dodds Series in Agricultural Science*, P. Langridge, Ed., Burleigh Dodds Science Publishing, 2017, pp. 263–294. doi:
10.19103/AS.2016.0004.19.

[44] S. Khanfri, M. Boulif, and R. Lahlali, “Yellow Rust (Puccinia striiformis): a Serious Threat to Wheat Production Worldwide,” *Notulae Scientia Biologicae*, vol. 10, no. 3, Art. no. 3, Sep. 2018, doi: 10.15835/nsb10310287.

[45] Y.-S. Kwak and D. M. Weller, “Take-all of Wheat and Natural Disease Suppression: A Review,” *Plant Pathol J*, vol. 29, no. 2, pp. 125–135, Jun. 2013, doi:
10.5423/PPJ.SI.07.2012.0112.

[46] R. P. Singh *et al.*, “Disease Impact on Wheat Yield Potential and Prospects of Genetic Control,” *Annual Review of Phytopathology*, vol. 54, no. 1, pp. 303–322, 2016, doi:
10.1146/annurev-phyto-080615-095835.

[47] Y. R. Mehta, “Disease Appraisal Scales,” in *Wheat Diseases and Their Management*, Y. R. Mehta, Ed., Cham: Springer International Publishing, 2014, pp. 249–252. doi: 10.1007/978-3-319-06465-9_9.

[48] E. Duveiller, H. Dubin, J. Reeves, and A.McNab, *Helminthosporium Blights of Wheat: Spot Blotch and Tan Spot*. 1997.

[49] S. A. H. Naqvi *et al.*, “Citrus Canker—Distribution, Taxonomy, Epidemiology, Disease Cycle, Pathogen Biology, Detection, and Management: A Critical Review and Future Research Agenda,” *Agronomy*, vol. 12, no. 5, p. 1075, Apr. 2022, doi: 10.3390/agronomy12051075.

[50] S. Iqbal, S. Iftikhar, and A. Rasool, “Assessment and molecular characterization of citrus canker causing pathotypes”.

[51] M. N. Aslam and T. Mukhtar, “Distributional spectrum of bacterial wilt of chili incited by Ralstonia solanacearum in Pakistan,” *Bragantia*, vol. 82, p. e20220196, 2023, doi: 10.1590/1678-4499-2022-0196.

[52] Z. Wang, W. Luo, S. Cheng, H. Zhang, J. Zong, and Z. Zhang, “Ralstonia solanacearum – A soil borne hidden enemy of plants: Research development in management strategies, their action mechanism and challenges,” *Front. Plant Sci.*, vol. 14, p. 1141902, Feb. 2023, doi: 10.3389/fpls.2023.1141902.

[53] D. P. Singh, “Plant Nutrition in the Management of Plant Diseases with Particular Reference to Wheat,” in *Recent Advances in the Diagnosis and Management of Plant Diseases*, L. P. Awasthi, Ed., New Delhi: Springer India, 2015, pp. 273–284. doi: 10.1007/978-81-322-2571-3_20.

[54] B. Bahri, S. J. A. Shah, S. Hussain, M. Leconte, J. Enjalbert, and C. De Vallavieille-Pope, “Genetic diversity of the wheat yellow rust population in Pakistan and its relationship with host resistance,” *Plant Pathology*, vol. 60, no. 4, pp. 649–660, Aug. 2011, doi: 10.1111/j.1365-3059.2010.02420.x.

[55] M. R. Khan *et al.*, “Diversity in *Puccinia striiformis* populations causing the 2013 yellow rust epidemics on major wheat cultivars of Pakistan,” *Mycologia*, vol. 112, no. 5, pp. 871– 879, Sep. 2020, doi: 10.1080/00275514.2020.1792263.

[56] R. Mumtaz, M. H. Maqsood, I. U. Haq, U. Shafi, Z. Mahmood, and M. Mumtaz, “Integrated digital image processing techniques and deep learning approaches for wheat stripe rust disease detection and grading,” *Decision Analytics Journal*, vol. 8, p. 100305, Sep. 2023, doi: 10.1016/j.dajour.2023.100305.

[57] W. Haider, A.-U. Rehman, N. M. Durrani, and S. U. Rehman, “A Generic Approach for Wheat Disease Classification and Verification Using Expert Opinion for Knowledge-Based Decisions,” *IEEE Access*, vol. 9, pp. 31104–31129, 2021, doi: 10.1109/ACCESS.2021.3058582.

[58] U. Shafi *et al.*, “Embedded AI for Wheat Yellow Rust Infection Type Classification,” *IEEE Access*, vol. 11, pp. 23726–23738, 2023, doi: 10.1109/ACCESS.2023.3254430.