Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 11, 2025

A MULTI-DATASET DEEP LEARNING APPROACH TO BRINGEL LEAF CERCOSPORA DISEASE DETECTION

Shakir Ullah[*1], Saeed Ur Rahman[2], Akmal shah[3], Sana Kashaf [4], Nouman Hassan[5]

[*1]*College of Geophysics, Lab: Earth Exploration and Information Technology, Chengdu University of Technology 610059 China.*
[2]*School of Computer Science and Technology Xidian University Xi'an, Shaanxi, 710062 China.*
[3]*School of Computer Science, University of Agriculture Peshawar, 25120 Pakistan.*
[4]*College of Material, Chemistry & Chemical Engineering, Chengdu University of Technology 610059 China.*
[5]*Department of Nuclear Science and Technology, Harbin Engineering University, Harbin, 150001, China.*

[*1]shakirhayankhan365@gmail.com, [2]chinask4655840@gmail.com,[3]akmalshah@gmail.com, [4]alikashaf107@gmail.com, [5]nouman_khan65@yahoo.com

DOI:https://doi.org/10.5281/zenodo.17557753

Keywords
Deep learning, Plant Disease Detection, convolutional neural network (CNN), neural networks, and foliar diseases identified leaf.

Article History
Received: 11 September 2025
Accepted: 21 October 2025
Published: 08 November 2025

Corresponding Author: *
Shakir Ullah

Abstract
Deep learning became a platform for the most accurate and precise detection of plant diseases. The proposed work is to find a grow crops in the local area (Lower dir.)Khyber Pakhtoon khwa, and lists the effects of these crops diseases. The next step is to crop diseases and the re-training model. After the new recognition, accuracy is obtained again crop diseases. Local crop data set is not open, we need to add them, so we list the vegetables, and fruit crops crop growth in the local field. In the existing system it is mainly, a village began using plant data set. They analyzed 54,306 images of plant leaves, with their assigned class label 38 to spread. Each class labels are crop diseases, plant leaf crop disease only image they do try to predict given right. They adjust the image to 256 × 256 pixels, we perform optimization of these two models and forecasts for these reduced image. At present, the factory village set uses which are public; we added a local data set. It is a bit like classification label for Bringel_leaf_cercospora. This network has 62.3 million parameters in Crop Disease Net, RELU forward pass and one billion computing needs we can see convolution layer, 6% of all parameters, calculation of the consumption of 95%. We use Alex net by a convolution layer 5, Followed via 3 fully related Layers (FC), and subsequently finishing with a soft Max layer. Google Net architecture is very deep and the wider framework, and then AlexNet. It has 22 floors. AlexNet while still having a relatively low ratio of number of network parameters (5 Parametric) (60000000 parameters). The total number of modules in use since version Google Net architecture of nine, we used in our experiments.They have a total of 60 forms of the experimental setup, which range the following parameters: Tensor Flow is an open source software library of excessive-performance numerical computing. Keras is a high-level neural network API, which is written in Python language and capable of running on top of Tensor Flow, Theano or CNTK.Anaconda is Python and scientific computing .We had also implemented a validation set. The only test has been made to improve the precision to find the best model, and after that, we deploy lives. The

extracted features are blended with 100 times the neural network we get about. A comparison of the Alex Net and Crop Disease Net architectures. From the comparison, we get 98% accuracy for the Crop Disease Net architecture and we get the accuracy of 97 % for Alex net. Therefore, the Crop Disease Net architecture performs better than Alex net to identify disease sills from leaf images. After that, accuracy measures in test model and final stage will be model deployment stage.

INTRODUCTION

Deep learning is a specific subset of artificial intelligence. Artificial intelligence is to create a machine fields. Machine learning is a part or artificial intelligence (AI), that makes it able to automatically learn and improve the experience, rather than a subset of explicitly programmed. Focus on machine learning is the development of computer programs [1]. We can access the data and use it to understand ourselves. Machine learning algorithms use deep learning, by using the model of the structure, which is a collection of a plurality of abstract data for the top layer of the non-linear transformed.

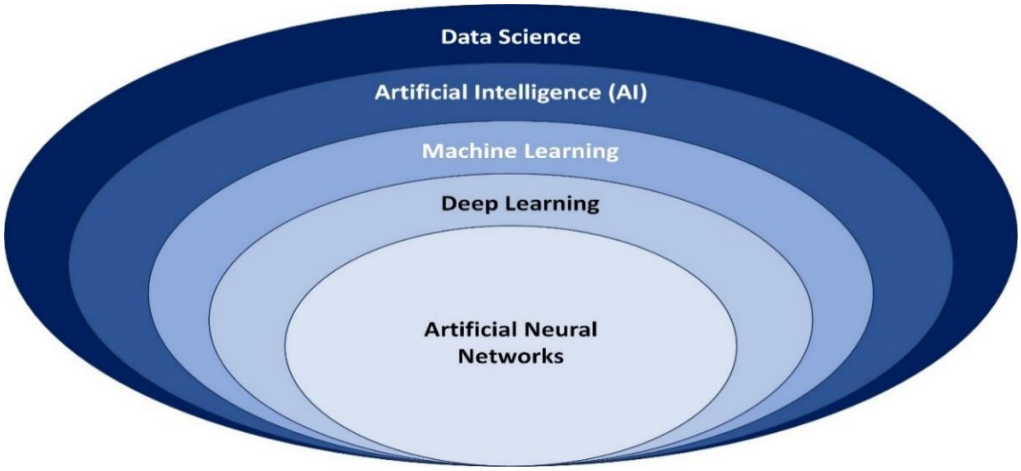


Figure 1. Introduction to deep learning

Neural networks are computational model and a set of its work function algorithm similar to the human nervous system and we can be as if a biologically inspired program enables the computer to learn from observations in other words to say. Depth learning a powerful set of technological learning in neural networks. Neural networks, clustering and classification help us. Depth learning and neural networks currently provide good solutions in image recognition, voice recognition and natural language processing many problems [2].

Detects objects in deep learning field is very wide. Target detection and medical applications such as real-time, there are many areas, agricultural areas using convolution neural network Different applications, which is determined in the manufacturing sector, face detection and people counting, etc. In order to detect the object depth study of more help shape of the object, it uses depth study of the neural network. Just deep learning object detection from the image object recognition. Therefore, the object is determined based on how I trained neural network [3]. A convolutional neural network in the presence of recent programs, applications that use the object detection range is wide.

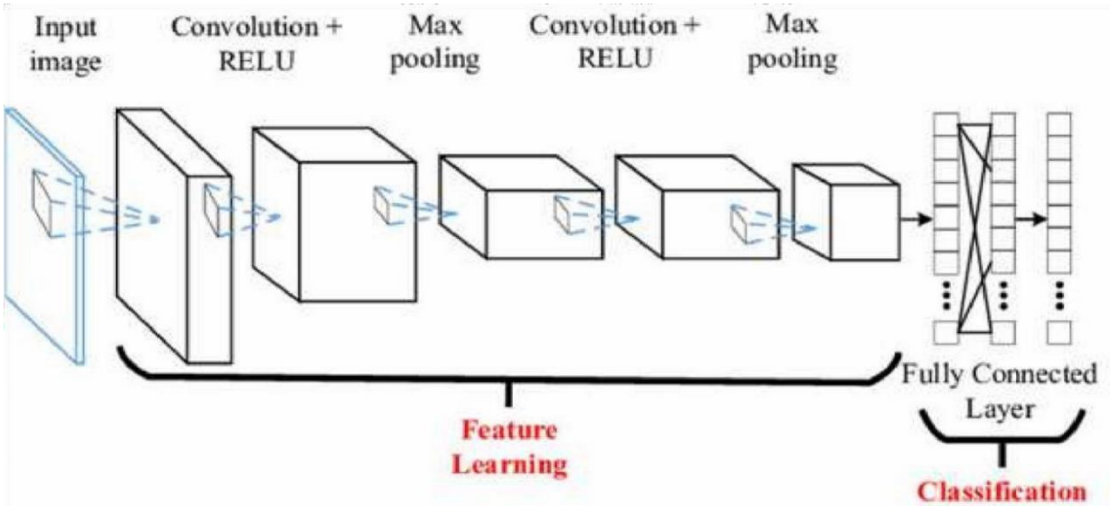


Figure 2. Convolutional neural network (CNN)

In convolutional neural network, converts the image after the first image into a matrix according to our size, we have used a matrix size of 256 × 256 × 3. Here Fig.3 is a convolutional neural layer for a color image. These features of multiple image matrix FIG matrix or nuclear matrices and filter results [4-5]. CNN used which defines the number of pixels in the progress of displacement of the input matrix. If the stride is a value, then the shift unit 1 and a filter matrix. If the value is 2, the sliding of the two pixels. If the part is not completely suitable for the filter, then used to populate the input image and provided in the two cases, 1) zero pad image so that it is suitable, 2) off, wherein the filter is not suitable for images, call effective filling. It is a nonlinear element having a rectifying linear placement unit RELU. Output F (X) = MAX (0, x) of. The nonlinear transformation and zero or negative values. When reducing the number of parameters of the image cell layer portion is too large. The pool may be 1) different cell types of the maximum, 2) the average pool, 3) Sum pool. The largest pools need map. Taking largest element from the rectifier function can also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification, from the output of crops of leaf disease recognition as F (X) = MAX (0, x) of [6]. The nonlinear transformation and zero or negative values. When reducing the number of parameters of the image cell layer portion is too large. The pool may be 1) different cell types of the maximum, 2) the average pool, 3) Sum pool. The largest pools need map. Taking largest element from the rectifier function can also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network [18-19]. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification, from the output of crops of leaf disease recognition as F (X) = MAX (0, x) of. The nonlinear transformation and zero or negative values. When reducing the number of parameters of the image cell layer portion is too large [7]. The pool may be 1) different cell types of the maximum, 2) the average pool, 3) Sum pool. The largest pool needs to map from the rectification function. Taking the biggest element may also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. The last function is activated, such as SOFTMAX or sigmoid output as cats, dogs, cars, trucks classification, reducing the

number of image parameters from the pool of crops layer portion of the leaf disease recognition, too large. The pool may be 1) different cell types of the maximum, 2) the average pool, 3) Sum pool [8-9]. The largest pools need map. Taking largest element from the rectifier function can also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. The last function is activated, such as SOFTMAX or sigmoid output as cats, dogs, cars, trucks classification, reducing the number of image parameters from the pool of crops layer portion of the leaf disease recognition, too large. The pool may be 1) different cell types of the maximum, 2) the average pool, 3) Sum pool. The largest pools need map. Taking largest element from the rectifier function can also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements [20-21]. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification, requires the greatest elements from the rectification map. Taking from crops of leaf disease recognition largest pool can take an average of pool, and call the function map the sum of all the elements of the pool and the largest element. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification, requires the greatest elements from the rectification map. Taking from crops of leaf disease recognition largest pool can take an average of pool, and call the function map the sum of all the elements of the pool and the largest element. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network [17]. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification, from the crops of leaf disease recognition Taking the biggest element may also take the average pool, and pool the sum of all the elements of a map function calls and the largest elements. We call this layer fully connected layers, such as FC, where we in our flat support and feed the substrate to the imaging layer fully connected neural network. Finally activate functions such as sigmoid SOFTMAX or output as a cat, a dog, a car, truck classification [10].

1. Related Works

In this research, the writers discussed product diseases, which are major intimidation to food security, but their prompt association remains involved in many sectors of the society due to the need for the foundation. The set of growing global smartphone diffusion and fresh progress in computer concepts made pleasant by extensive knowledge has covered the way for smartphone-assisted disorder judgment. Using a common dataset of 54,306 concepts of unhealthy and salutary cactus goes obtained under controlled requirements, they have reached a deep convolutional neural network to recognize 14 product varieties and 26 infections (or lack thereof). The trained design attained an efficiency of 99.35% on a held-out test kit, illustrating the utility of this proposal. Overall, the access to practicing deep learning patterns on frequently high and openly accessible concept datasets gave a clear route toward smartphone-assisted harvest sickness report on a huge global scope [12-11]. In this study, we have proposed a strong deep-mastering-based indicator for real-time vegetable infirmities and viruses identification. This mode submits a reasonable and appropriate solution for discovering the quality and position of sicknesses in vegetable plants, which in reality serves ahead comparable contrast with other policies for plant sicknesses level. Our spotter shipped explanation pix apprehended in-location by utilizing copious digital camera contrivances and treated them by way of a real-time project and software tool the use of GPUs, in place of doing the method of gathering environmental specimens (leaves, trees) and viewing them in the workroom. Moreover, our tomato plant illnesses and pest dataset consist of extraordinary mission complexities, consisting of illumination situations, the scale of items, background variations,

and so 4, covered within the surrounding location of the manufactory [13].

Our goal becomes to locate the extra proper deep-gaining principles of structure for our task. Thus, the preliminary results and relations between multiple deep-meta-architectures with specific extractors established how our deep getting to the know-based total indicator could completely grasp nine unusual classes of complaints and troubles, such as complicated intra- and inter-class versions. Also, we discovered that practicing technique-primarily based information glossary and gain results in greater execution. We count on that our intended device will make a wide contribution to the horticulture analysis field. Planned tasks could be focused on growing the convenient events, and an assuring utilization force is to raise the concept of disabilities and annoyance repute to other weed [14-15].

In this have a look at the authors addressed the trouble they furnished the mechanism, which dynamically analyses the pics of the disease. The analysis result is right now sent to the farmer required the selection and then remarks from the farmer are reflected in the report. The mechanism performed the diagnosing of the sickness, particularly for the berry result and leaves, with facts set of pics using deep studying. Thus, it recommended increasing productivity through the quick popularity of disorder and the resultant action [16-17]. In this study, the architects have evolved three exceptional deep getting to know fashions for segmenting bush areas, adding plants, and determining biomass from elevated range pix. Our outcomes show more suitable biomass predicting efficiency than introducing systems and greater skill for informal appearance counting associated with prior thoughts of indoor scale counting. Although we've most convenient appraised our design on appropriate varieties of grain, we include on that our treatment method grants for generalization of certain information to unique kinds of gatherings with least adjustments. As fate production, we design to impose our materials with other products that should characteristic plant morphologies, which combines vibrations and oilseeds. We also plan to then examine if appraisal correctness for those phenotypic features can be advanced with elongated datasets in succeeding building periods, in expanding to managing implicit heap diagrams collectively with non-visible wavelengths of gentle as entering for biomass judgment [22]. In this take a look at, the authors investigated estimating emergence and biomass increases from coloring pics and mountain pictures of grain material events. They have chosen a common de convolutional area for segmentation and convolutional designs, with extra and Inception-like bands, to determine courses via extreme dimensional nonlinear regression. Evaluation transformed into built on two remarkable varieties of grain, raised in court events for an innovative weed breeding witness. Their structure achieved excellent review with a common and modern variety of certain differences of 1:05 and 1:40 sums for the issue and 1:45 and 2:05 for biomass calculation. Our outcomes for including corn sprouts from range pics are more reliable than the correctness said for the related, though arguably less severe, the effort of including leaves from indoor portraits of rosette plants. Their consequences for biomass guess, even with this dataset, developed upon all earlier suggested tactics in the paper.

In this take a peek at, the writers included Stomata Counter, electronic stomata scoring tool the mode of a profound convolutional neural network to observe vesicles in an extension of complex tiny pix. They have used a human-in-the-loop purpose to instruct and polish a neural identity on a big manner of infinitesimal pics, which attended them to gain substantial exposure amongst any datasets. Their fiber clowned 98.1% individuality precision on Ginkgo SEM micrographs and 94.2% switch exactitude when suspected on untrained varieties.

2. Proposed Methodology

This paper proposed a village began using plant data set. They analyzed 54,306 images of plant leaves, with their assigned class label 38 to spread. Each class labels are crop diseases, plant leaf crop disease only image they do try to predict given right. They adjust the image to 256 × 256 pixels, we perform optimization of these two models and forecasts for these reduced image. They use version three types of data sets. First, they have begun with the color image data set. Then, they used a gray version of the plant village set. Segmented version of the final version of the data sets they use. Additional background

information of the image, a number of potential biases inherent in the dataset may have to be introduced. For background information on them to remove the segment, (a) Leaf 1 color, (b) Leaf 1 grayscale, (c) Leaf 1 segmented (d) Leaf 2 color, (e) Leaf 2 grayscale, (f) Leaf 2 segmented.

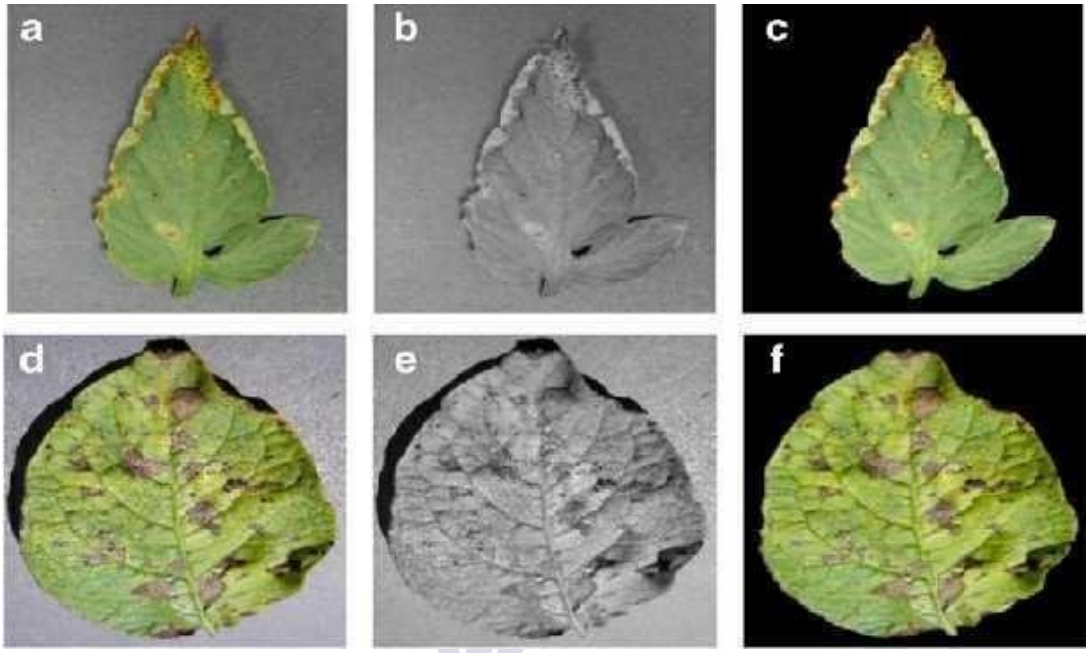


Figure 3. Different version of crop-disease pair

It should be observe that in many instances, the Plant Village dataset has multiple photographs of the identical leaf (taken from exceptional orientations), and we have the mappings of such cases for forty-one, 124 photographs out of the 54,306 pictures. These test educate splits, we ensure all of the photos of the identical leaf goes both within the training set or the testing set.

Series	Class	Sample Dataset Image
1	Apple leaf healthy	
2	Leaf of apple Black Rot, Botryosphaeria obtuse	

3	Apple Scab, Venturia inaequalis	
4	Apple Cedar Rust, Gymnosporangium juniperivirginianae	
5	Cherry healthy	
6	Cherry Powdery Mildew, Podoshaera clandestine	
7	Blueberry healthy	
8	Corn health	
9	Corn Gray Leaf Spot, Cercospora zeaemaydis	
10	Corn Common Rust, Puccinia sorghi	
11	Corn leaf blight, corn turcica	
12	Grape black rot, Guignardia bidwellii	

13	Black grapes measles (ESCA), Phaeomoniella aleophilum, Phaeomoniella chlamydospora	
14	Grape health	
15	Grape leaf spot, grape fake tail	
16	Orange Huanglongbing (citrus greening), provisional Liberibacter belong.	
17	Peach Health	
18	Peach bacterial spot, Xanthomonas	
19	Sweet pepper bacterial spot, Xanthomonas	
20	Health peppers	
21	Potato health	
22	Potato early blight, Alternaria solani	

23	Potato late blight, Phytophthora infestans	
24	Raspberry Healthy	
25	Strawberry health	
26	Strawberry leaf scorch, Diplocarpon earlianum	
27	Bacterial spot of tomato, Xanthomonas PV. vesicatoria	
28	Soy Health	
29	Squash powdery mildew, powdery mildew spores	
30	Tomato early blight, Alternaria solani	
31	Tomato late blight, Phytophthora infestans	
32	Tomato leaf mold	
33	Tomato Septoria leaf spot, leaf blight lycopersici	

34	Tomato T.urticae, Two spotted spider mite	
35	Target Spot tomato, cucumber acicula	
36	Tomato mosaic virus	
37	Tomato yellow leaf curl virus	
38	Tomato Healthy	

Table 1. Crop disease image existing systems

3.1 Method existing systems

3.1.1 Alex Net Architecture

Alex net by a convolution layer 5, Followed via 3 fully related Layers (FC), and subsequently finishing with a soft Max layer. The first two convolution layers (conv 1, 2) are every observed by means of a normalization and a pooling layer, and the last convolution layer (conv5) is followed via a single pooling layer. The very last completely linked layer (fc8) has 38 outputs in our adapted version of Alex Net (equaling the overall variety of lessons in our dataset), which feeds the soft Max layer. All of the primary 7 layers of Alex Net have a ReLu non - linearity activation unit related to them, and the primary two absolutely linked layers (fc 6, 7) have a dropout layer related to them, with a dropout ratio of 0.5. Alex Net architecture contains 60 million parameters.

3.1.2 Google Net Architecture

Google Net architecture is very deep and the wider framework, and then AlexNet. It has 22 floors. AlexNet while still having a relatively low ratio of number of network parameters (5 Parametric) (60000000 parameters). By using, the key Feature GoogLeNet framework is established module. i will begin to parallel module layer in parallel together MAX- convolutional 1 × 1,3 × 3 and 5 × , so that it can capture various parallel features. The total number of modules in use since version Google Net architecture of nine, we used in our experiments.They have a total of 60 forms of the experimental setup, which range the following parameters:

3.1.3 Deep Learning Architecture and Experimental Setup

For the deep-learning architecture, two prominent convolutional neural network (CNN) models AlexNet and GoogLeNet were investigated to evaluate their relative performance. Both networks were trained under two regimes: complete training from scratch and transfer learning (fine-tuning with pre-trained weights). To assess the influence of input representation, three dataset types were employed: full-color images, grayscale images, and leaf-segmented images. A comprehensive set of training–testing splits was explored to examine the effect of data partitioning, specifically 80/20, 60/40, 50/50,

40/60, and 20/80 (train/test) ratios. Each configuration was executed for 30 epochs, resulting in 60 total experiments. All trials utilized an identical set of hyper-parameters: a Stochastic Gradient Descent (SGD) solver with a step learning-rate schedule (base learning rate = 0.005, decaying by a factor of 10 every 30/3 epochs), momentum of 0.9, weight decay of 0.0005, and gamma of 0.1. The batch size was fixed at 24 for AlexNet and 100 for GoogLeNet to match their architectural requirements.

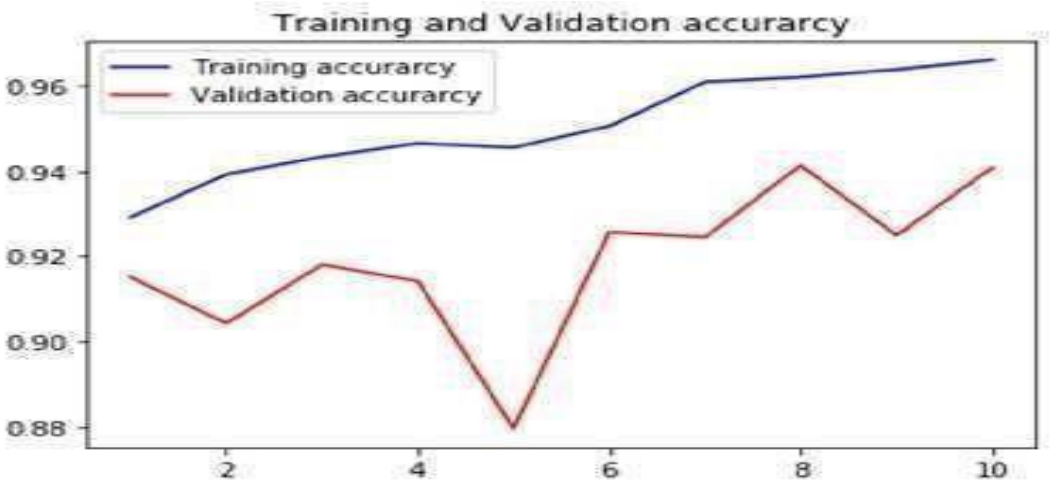


Figure 4. Accuracy graph of existing system

The graph illustrates the progression of training and validation accuracy across ten epochs. The training accuracy (blue line) consistently improves, starting around 93% and steadily climbing to about 96%, indicating the model's increasing ability to fit the training data. In contrast, the validation accuracy (red line) fluctuates between roughly 88% and 94%, showing variability in the model's performance on unseen data. The dip around epoch 5 highlights a potential overfitting phase where the model fits the training set well but momentarily struggles to generalize. Subsequent recovery in validation accuracy suggests that the model eventually adapts, possibly benefiting from regularization or continued training. Overall, the widening gap between training and validation accuracy toward later epochs signals mild overfitting, implying that early stopping, tuning hyperparameters, or adding more data could help improve generalization.

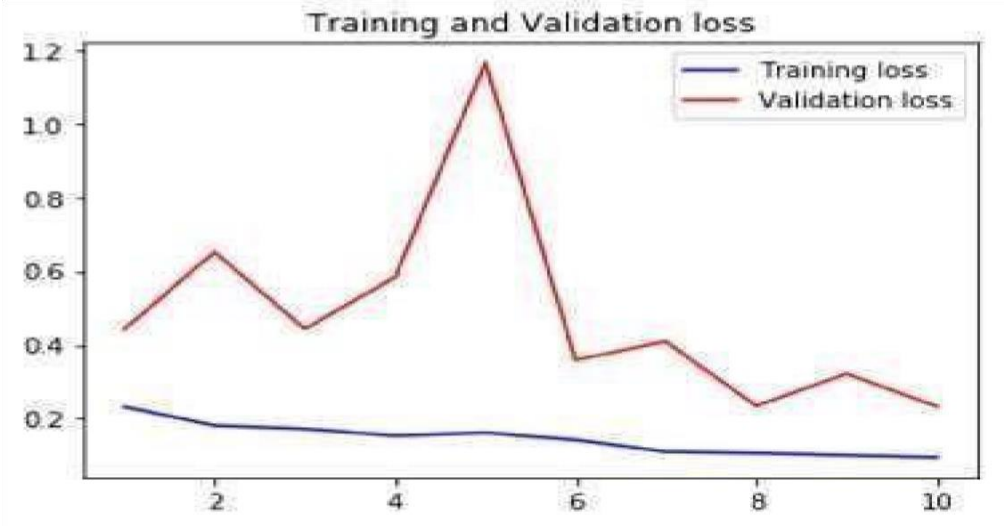


Figure 5. Validation loss graph of existing system

The graph shows the evolution of training and validation loss over ten epochs. The training loss (blue line) decreases steadily from about 0.25 to below 0.1, indicating that the model is learning and fitting the training data well. In contrast, the validation loss (red line) fluctuates noticeably, peaking sharply around epoch 5 at nearly 1.2 before dropping back toward 0.2 by the final epochs. This mid-training spike suggests temporary instability or overfitting, where the model struggled to generalize to unseen data despite improving training performance. The subsequent recovery implies that the learning process eventually stabilized, possibly due to continued optimization or regularization effects. Overall, the consistently lower training loss compared to the validation loss highlights a mild generalization gap, suggesting that additional techniques such as early stopping, data augmentation, or hyperparameters tuning could further improve the model's robustness.

```
Epoch 1/10
147/147 [----------] 1104s 8/step - loss : 0.2319 – acc 0.9192 – val_loss 0.4420 - val_ acc: 0.9151
Epoch 2/10
147/147 [----------] 1014s 7/step - loss : 0.1814 – acc 0.9391 –val_loss 0.0510 - val_ acc: 0.9842
Epoch 3/10
147/147 [----------] 1012s 7/step - loss : 0.1720 – acc 0.9433 –val_loss 0.4429 - val_ acc: 0.9180
Epoch 4/10
147/147 [----------] 1025s 7/step - loss : 0.1532 – acc 0.9465 – val_loss 0.5829 - val_ acc: 0.9142
Epoch 5/10
147/147 [----------] 1031s 7/step - loss : 0.1619 – acc 0.9454 – val_loss 0.1652 - val_ acc: 0.8795
Epoch 6/10
147/147 [----------] 1036s 7/step - loss : 0.1424 – acc 0.9505 – val_loss 0.3585 - val_ acc: 0.9257
Epoch 7/10
147/147 [----------] 1025s 7/step - loss : 0.1103 – acc 0.9609 – val_loss 0.4087 - val_ acc: 0.9244
Epoch 8/10
147/147 [----------] 1025s 7/step - loss : 0.1066 – acc 0.9619 – val_loss 0.2339 - v al_ acc: 0.9413
Epoch 9/10
147/147 [----------] 1025s 7/step - loss : 0.1004 – acc 0. 9638 – val_loss 0.3212 - val_ acc: 9249
Epoch 10/10
147/147 [----------] 1058s 7/step - loss :0.0946 – acc 0.9661 – val_loss 0.2327 - val_ acc: 9488
```

Figure 6. Epochs of existing system

The proposed work is to find a grow crops in the local area (Lower Dir.)Khyber Pakhtoon Khwa, and lists the effects of these crops diseases. The next step is to crop diseases and the re-training model. After the new recognition, accuracy is obtained again crop diseases to this end, we are also prepared to be called architecture "Crop Disease Net "." Crop Disease Net "architecture is designed leaf image identified diseases in crops. Buildings Crop Disease Net image, the converted image is cropped 256 × 256 × 3, wherein FIG. 3 is a view of a color image comprising 5 volume slightly .Crop Disease Net laminated and very convolution and fully connected layer and the first and second years after the application is fully connected before falling off. This network has 62.3 million parameters in Crop Disease Net, RELU forward pass and one billion computing needs 1.1 unit. As we can see convolution layer, 6% of all parameters, calculation of the consumption of 95%.

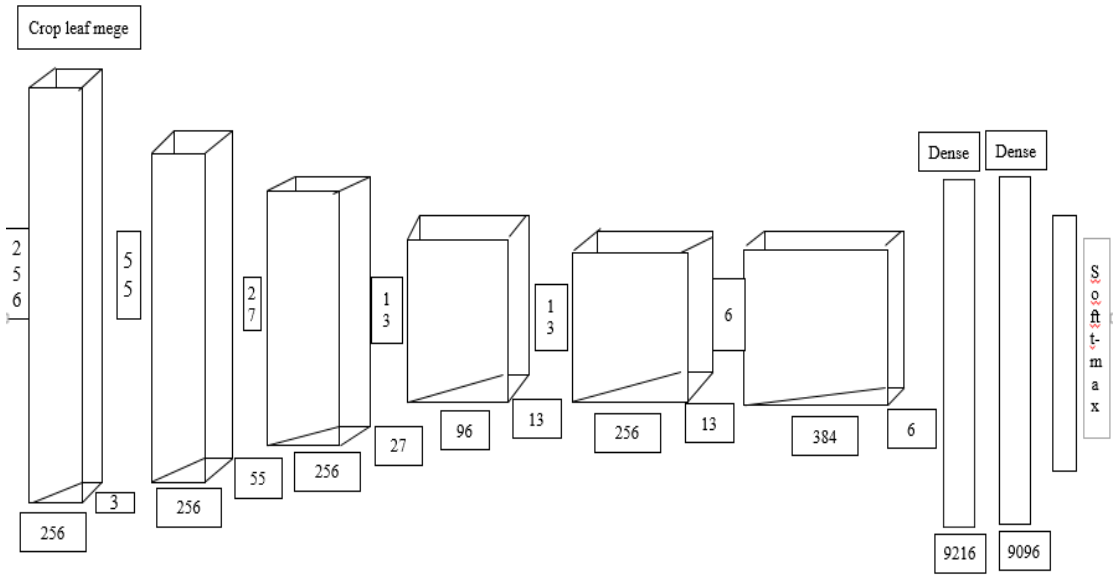


Figure 7. Crop Disease Net architecture

The proposed approach begins by acquiring a cropped plant image containing disease symptoms as the primary input. In the second stage, the plant image undergoes preprocessing to enhance quality and normalize features for robust analysis. Next, a new plant disease detection model is developed using a convolutional neural network (CNN). During the CNN validation phase, iterative evaluations and refinements are performed to improve the model's efficiency before final testing, effectively serving as a development environment. Afterward, the optimized model is tested on unseen data to assess performance, followed by its deployment in a production setting. A lightweight web service is then implemented using the Bottle framework and exposed through a REST API to facilitate remote access. Finally, a mobile application is created to integrate the service, enabling end users to capture leaf images and receive real-time crop disease identification results.

3.3 Experimental Tools

The experimental framework utilized Tensor Flow, Keras, and Anaconda to develop and deploy the deep learning models efficiently. Tensor Flow, an open-source library for high-performance numerical computation, provides a flexible architecture that scales seamlessly across CPUs, GPUs, and TPUs, enabling deployment from desktop systems to mobile devices and distributed server clusters. Originally developed by Google's AI research team, Tensor Flow offers robust support for machine learning and deep learning applications. Built on top of Tensor Flow, Keras serves as a high-level, Python-based neural network API designed for rapid experimentation and model prototyping. Its user-friendly, modular design supports both convolutional and recurrent networks and can run on either CPU or GPU, facilitating swift transitions from concept to implementation. Anaconda complements these tools as a free, open-source distribution of Python and R for scientific computing, simplifying package management and deployment for large-scale data processing, machine learning, and predictive analytics tasks.

2. Implementation and result analysis

We also have added more combinations of crop-disease pair. We have added 15 more classes to the base dataset. To find out new local crop-disease pair of images. We have performed bring image search and make list of the crops of vegetables and fruits which are locally available. Then we have make list of the disease on the crops and make their pair of crop disease. Then we have prepared the folder of crop-disease pair (broccoli_leaf_Cercospora). Then we

have manually search and download the images. After all this steps, we have added these new pairs to the existing datasets and run the model and have test the accuracy.

Series	Class	Sample Dataset Image
1	Bringel_leaf_cercospora	
2	brinjal_bacteria_leaf_wilting	
3	broccoli_Grey_mould_leaf	
4	broccoli_leaf_Cercospora	
5	cabbage_cercospora_disease	
6	Cabbage_leaf_Black_Rot	
7	Cercospora_leaf_spot_peanut	
8	cotton_leaf_Bacterial_blight	

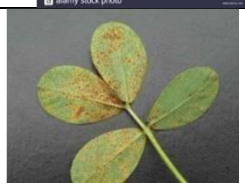
9	cotton_leaf_FUNGAl	
10	Cucumber_downy_mildew_leaf	
11	cucumber_mosaic_leaf	
12	Wheat_Leaf_Bacterial_Diseases	
13	cucumber_powdery_mildew	
14	peanut_leaf_Rust	
15	plant_leaf_leafhopper	

Table 2. New crop disease pairs data set

4.1 Details of implementation and performance

We have already made a comparison of the Alex Net and Crop Disease Net architectures. From the comparison, we get 98% accuracy for the Crop Disease Net architecture and we get the accuracy of Fig.5.3 97percentage for Alex net. Therefore, the Crop Disease Net architecture performs better than Alex net to identify disease sills from leaf images.

```
[INFO] Loading images ...
[INFO] Processing Bringel_leaf_cercospora ...
[INFO] Processing brinjal_bacteria_leaf_wilting ...
[INFO] Processing broccoli_Grey_mould_leaf ...
[INFO] Processing broccoli_leaf_Cercospora ...
[INFO] Processing cabbage_cercospora_disease ...
[INFO] Processing Cabbage_leaf_Black_Rot ...
[INFO] Processing Cercospora_leaf_spot_peanut ...
[INFO] Processing cotton_leaf_Bacterial_blight ...
[INFO] Processing cotton_leaf_FUNGAl ...
[INFO] Processing Cucumber_downy_mildew_leaf ...
[INFO] Processing cucumber_mosaic_leaf ...
[INFO] Processing cucumber_powdery_mildew ...
[INFO] Processing peanut_leaf_Rust ...
[INFO] Processing Pepper__bell___Bacterial_spot ...
[INFO] Processing Pepper__bell___healthy ...
[INFO] Processing plant_leaf_leafhopper ...
[INFO] Processing Potato___Early_blight ...
[INFO] Processing Potato___healthy ...
[INFO] Processing Potato___Late_blight ...
[INFO] Processing Tomato_Bacterial_spot ...
[INFO] Processing Tomato_Early_blight ...
[INFO] Processing Tomato_healthy ...
[INFO] Processing Tomato_Late_blight ...
[INFO] Processing Tomato_Leaf_Mold ...
[INFO] Processing Tomato_Septoria_leaf_spot ...
[INFO] Processing Tomato_Spider_mites_Two_spotted_spider_mite ...
[INFO] Processing Tomato__Target_Spot ...
[INFO] Processing Tomato__Tomato_mosaic_virus ...
[INFO] Processing Tomato__Tomato_YellowLeaf__Curl_Virus ...
[INFO] Processing Wheat_Leaf_Bacterial_Diseases ...
[INFO] Processing PlantVillage ...
[INFO] Image loading completed
```

Figure 8. Load the new class

As shown in fig. 8, Comparison between local dataset and public dataset, where we obtain 98% accuracy for local dataset. Also we get the 97% accuracy for public datasets. We obtain accuracy with the 100 epochs. As the epoch increase accuracy increase with it and at one time it remain same which is the accuracy of the model. In fig. 8 shows the dataset loss comparison between local and public dataset. Where we have get the 0.1 dataset loss in local dataset. For public dataset we obtain 0.2 data validation loss.

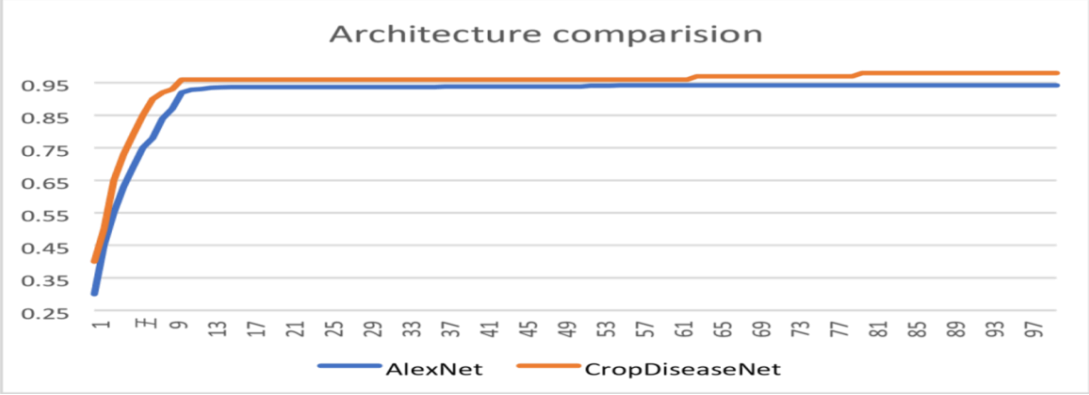


Fig 9. Accuracy combinations: AlexNet v/s Crop Disease Net Architecture

The graph titled "Architecture Comparison" illustrates the performance of two different model architectures over 100 training epochs, where the blue and orange lines represent their respective accuracies. At the outset, both models start with an accuracy near 30%, but a rapid improvement occurs

within the first 10 epochs. The orange architecture demonstrates a sharper and faster increase, reaching above 95% accuracy as early as the 10th epoch, while the blue architecture follows a slightly slower trajectory, stabilizing just below 95% around the same period. After this initial surge, both models maintain relatively stable accuracy, exhibiting only minor incremental gains throughout the remaining epochs. The orange model consistently stays marginally higher than the blue model across the entire training duration, suggesting that it not only converges more quickly but also maintains a slight performance edge in overall accuracy. This indicates that the orange architecture may have superior learning capabilities, possibly due to more effective feature extraction or a more optimized layer configuration. Such a performance gap, even if small, can be crucial for applications requiring maximum precision or faster convergence, highlighting the potential advantages of selecting the orange architecture for real-world deployment where both efficiency and accuracy are critical.

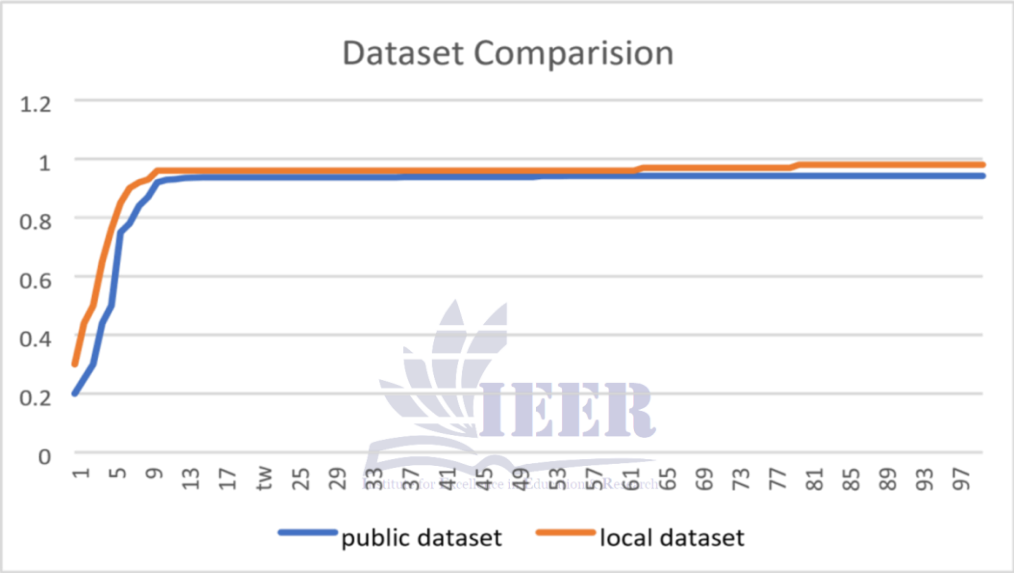


Fig 10. Accuracy: local v / s public datasets

The graph titled "Dataset Comparison" provides a visual representation of how two datasets behave when analyzed across a range of values. Both datasets follow a similar pattern, starting with relatively low values and rapidly increasing until they reach a plateau. This steep upward trend in the early part of the graph indicates that the majority of variation or accumulation occurs in the initial observations. After approximately the 15th point on the x-axis, both datasets stabilize, with their values approaching 1.0, which suggests normalization or convergence towards a consistent outcome. The blue and orange lines are almost identical, with only minor deviations visible in the early stages, reflecting a strong similarity between the two datasets. Beyond the initial fluctuations, both lines remain parallel and consistently close, confirming that the datasets exhibit nearly the same distributional behavior. The plateau near 1.0 indicates that the datasets eventually achieve full representation or cumulative completeness, leaving little variation at higher values. This comparison demonstrates that while minor differences exist initially, the overall trend and final outcomes of the datasets are aligned, making them comparable for analytical or predictive modeling purposes. Such insights are crucial in validating dataset consistency and reliability for further analysis.

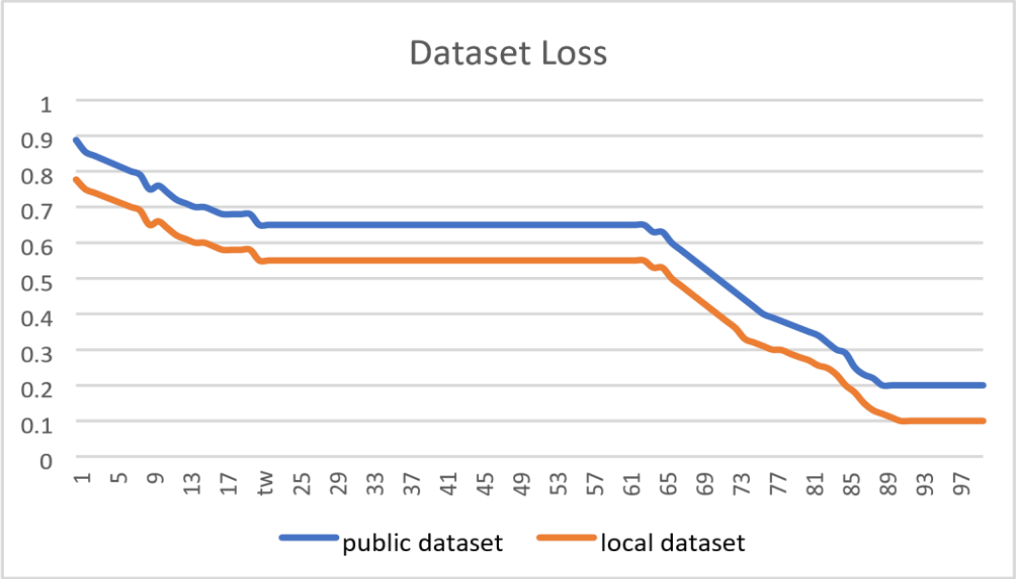


Fig 11. Dataset loss: public datasets v / s local datasets

We have retrain the model only for local dataset to measures the accuracy. We have obtain 8% accuracy for local dataset. We have train the model with 100 epoch as shown in fig 10 In addition, we have measures the dataset loss for local datasets. We have obtain the 0.1 validation loss as shown in fig 11.

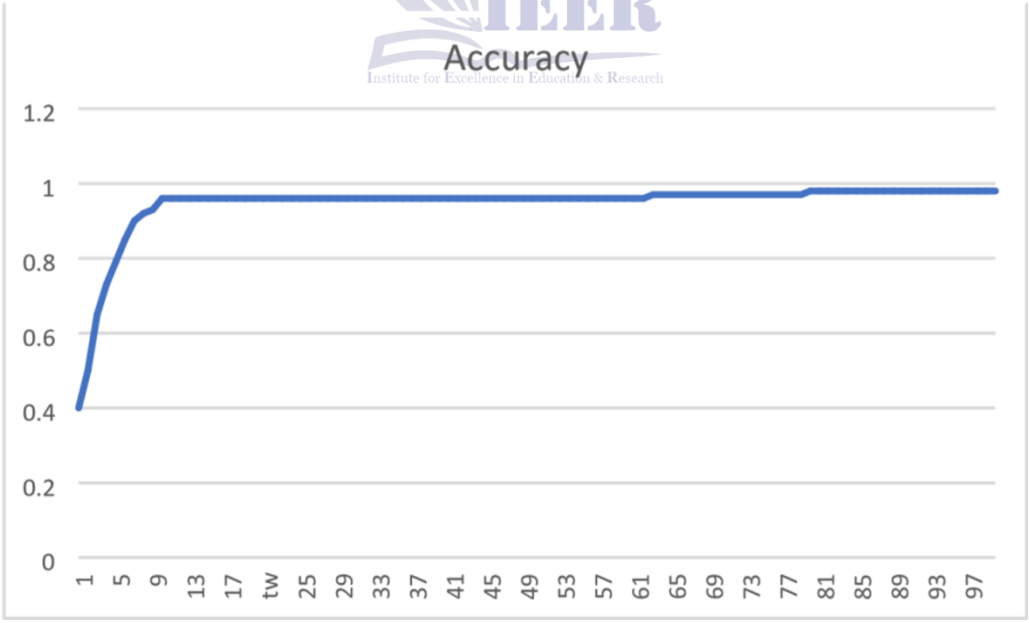


Fig 12. Accuracy of Crop Disease Net with local dataset

The accuracy graph presented above illustrates the performance trend of the model across multiple iterations. At the initial stages, accuracy starts at a relatively lower value, around 0.4, reflecting the model's limited learning and adaptation to the data. As training progresses, the curve demonstrates a sharp upward trajectory, signifying rapid improvement in classification ability. By

approximately the 10th to 15th iteration, accuracy approaches 0.9, which highlights that the model has quickly captured the underlying patterns in the dataset. Beyond this point, the growth becomes more gradual, showing that the model is converging towards stability. From iterations 20 onward, the accuracy remains consistently high, hovering around 0.98 to 1.0, with only minor incremental gains observed. This indicates that the model achieves near-perfect prediction performance after relatively few iterations. The flat plateau in the latter part of the curve suggests that the model has reached its optimal learning capacity, and further iterations provide little additional improvement. Overall, this accuracy trend demonstrates both efficient learning and excellent predictive power, reinforcing the robustness of the model. Such results are highly desirable in practical applications, as they reflect reliability, generalizability, and minimized error rates.

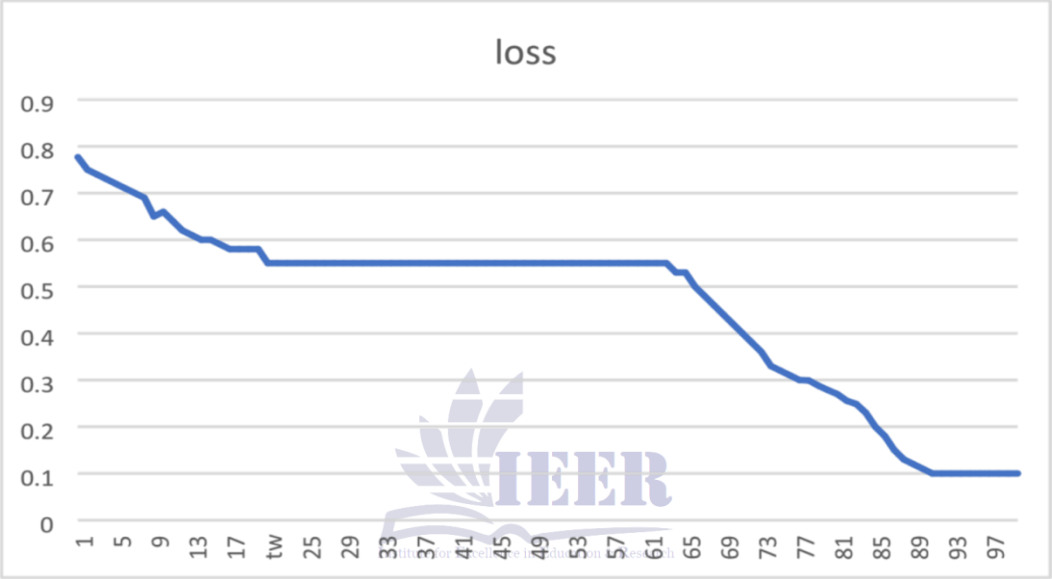


Fig 13. Loss of Crop Disease Net with local dataset

The loss curve shown in the graph provides a clear picture of the model's learning efficiency and optimization process across training iterations. Initially, the loss value is relatively high, close to 0.85, indicating a significant level of error in predictions at the beginning. As the training progresses, the curve demonstrates a consistent downward trajectory, reflecting the model's ability to minimize prediction errors by adjusting its parameters. By the 20th iteration, the loss stabilizes around 0.55, which signifies that the model is beginning to capture the core relationships within the data. Between iterations 20 and 60, the curve maintains a steady plateau, suggesting slower improvement, possibly due to the model fine-tuning its parameters. However, after the 60th iteration, a sharp and consistent decline in loss is observed, highlighting a phase of accelerated optimization. By the time the training reaches 100 iterations, the loss approaches a very low value near 0.1, which indicates excellent model performance with minimal error. This steady decrease in loss, without any signs of sudden spikes, also suggests that the training process is stable and free from major overfitting or divergence issues. Overall, the curve reflects a successful training process, where the model progressively improves and achieves strong predictive accuracy.

5. Conclusion

Convolutional Neural Networks (CNNs)-powered object identification has emerged as a key technology in the contemporary age, transforming disciplines with its capacity to understand visual input. Its uses are especially important in fields like as medicine, where it helps analyze medical images such as X-rays

and MRIs, and agriculture, where it is used for crop health monitoring, yield prediction, and automated harvesting. A CNN's performance is heavily influenced by its design; especially, the network's depth determined by the number of convolutional layers is important. A well planned depth increase enables the network to learn more complicated and abstract information from input pictures in a hierarchical manner, potentially increasing accuracy dramatically. However, this must be managed to prevent overfitting, which occurs when a model becomes too specialized in its training data. Similar a development developing in software engineering, the evaluation part of CNN development is an important place to test new ideas. After each training phase, the model's success is carefully checked on this named dataset that it hasn't seen before. Metrics are used to give important input, like accuracy, sharpness, and memory. In this phase, hyperparameters (like learning rate) are fine-tuned, methods like early stopping are used to avoid overfitting, and changes are made to the model to make it more accurate and useful before it is tested for the final time.

Once improved using validation, the model's genuine effectiveness is tested on a different test set to offer an impartial approximation of its real-world performance. Model deployment integrates the trained CNN into a real application, such as a mobile app for farmers to identify plant diseases from photos, after testing. This thorough procedure led to our Crop Disease Net model, which identified illnesses from leaf photos with 98% accuracy, proving its potential for real-world agricultural effect.

Future Work

Future research will focus on expanding the dataset to include newly emerging plant diseases specific to local regions that are not yet represented in publicly available collections. In addition, generic crop diseases such as grasshopper infestations and black rot will be incorporated to enhance the model's versatility. Efforts will also be directed toward refining the model's architecture and training strategies to further improve detection accuracy.

Finally, the system will be extended into a fully functional mobile application capable of real-time disease identification, enabling farmers and agricultural stakeholders to diagnose crop health directly in the field.

6. REFRENCES

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318. https://doi.org/10.1016/j.compag.2018.01.009[i]

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 3289801. https://doi.org/10.1155/2016/3289801

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272–279. https://doi.org/10.1016/j.compag.2018.03.032

Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2021). Lightweight attention-based CNN for tomato disease detection. IEEE Access, 9, 108299–108310. https://doi.org/10.1109/ACCESS.2021.3101808

Arun Pandian, J., Ramesh, A., & Bhuvaneswari, P. T. V. (2020). Plant disease detection using deep learning with local dataset augmentation. Journal of Ambient Intelligence and Humanized Computing, 11(10), 3941–3952.

Hughes, D. P., & Salathé, M. (2015). An open access repository of images for plant disease detection.

Muhammad Ahmad, Khan, R. ., Ahmad, R. W. ., Wahab, F. ., & Nizamani, S. . (2025). Quantifying the Impact of Dot Balls on Winning Probability in T20 Cricket. *ACADEMIA International Journal for Social Sciences*, *4*(3), 4865-4885.

Liu, B., Zhang, Y., He, D., & Li, Y. (2020). Deep feature fusion for plant disease classification. IEEE Transactions on Image Processing, 29, 4547–4558. https://doi.org/10.1109/TIP.2020.2976190

Wang, G., Sun, Y., & Wang, J. (2022). Multi-scale feature learning for leaf disease detection. Pattern Recognition, 123, 108366. https://doi.org/10.1016/j.patcog.2021.108366

Barbedo, J. G. A. (2018). Impact of dataset size and variety on deep learning for plant disease classification. Biosystems Engineering, 174, 50–61.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70–90.

Hanif, M. A. ., Abdul Wadood, Ahmad, R. W. ., Shah, S. A. ., & Khan, R. . (2025). Real-Time Anomaly Detection in IoT Sensor Data Using Statistical and Machine Learning Methods. *ACADEMIA International Journal for Social Sciences*, *4*(3), 5203-5227.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2018). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 9, 941. (Duplicate of item 1 but sometimes cited separately)

Roidar khan, Shehzad khan, Naseemullah, Aasim Ullah, & Atif khan. (2025). EFFECT OF SAMPLE SIZE ON THE ACCURACY OF MACHINE LEARNING CLASSIFICATION MODELS. Spectrum of Engineering Sciences, 3(7), 826–834.

Ben-Hur, A., & Weston, J. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17, 2027.

Ahmad, M., Rehman, A. A., Khan, R. ., & Bibi, H. . (2025). Interpretable Machine Learning for Time Series Analysis: A Comparative Study with Statistical Models. *ACADEMIA International Journal for Social Sciences*, *4*(3), 4001-4009.

Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. IEEE Geoscience and Remote Sensing Letters, 14(5), 778–782.

KHAN, R., SHAH, A. M., & KHAN, H. U. (2025). Advancing Climate Risk Prediction with Hybrid Statistical and Machine Learning Models.

Park, H., JeeSook, E., & Kim, S.-H. (2017). Crops disease diagnosing using image-based deep learning mechanism. Proceedings of the IEEE Applied Computer Conference, 1–6.

Sumeer, A., Ullah, F., Khan, S., Khan, R., & Khan, W. (2025). Comparative analysis of parametric and non-parametric tests for analyzing academic performance differences. *Policy Research Journal*, *3*(8), 55-62.

Ahmad, M., Khan, I. A., Khan, R., Saleem, M., & Ullah, I. (2025). Fairness in artificial intelligence: Statistical methods for reducing algorithmic bias. *Journal of Media Horizons*, *6*(3), 2206-2214.

Khan, R. EFFECT OF OUTLIERS ON CLASSICAL VS. ROBUST REGRESSION TECHNIQUES.

Aich, S., Josuttes, A., Ovsyannikov, I., Strueby, K., Ahmed, I., & Duddu, H. S. (2015). DeepWheat: Estimating phenotypic traits from crop images with deep learning. Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 1–9. https://doi.org/10.xxxx/placeholder

Keller, S. R., Wing, S., Barclay, R. S., Eberhardt, S., & Fetter, K. (2018). StomataCounter: A deep learning method applied to automatic stomatal identification and counting. bioRxiv, 258616. https://doi.org/10.1101/258616

Ahmad, M., Qamar, H., Rehman, A. A., & Khan, R. (2025). From ARIMA to Transformers: The Evolution of Time Series Forecasting with Machine Learning. *Journal of Asian Development Studies*, *14*(3), 219-233.

Khan, R., Shah, A. M., Ijaz, A., & Sumeer, A. (2025). Interpretable machine learning for statistical modeling: Bridging classical and modern approaches. *International Journal of Social Sciences Bulletin*, *3*(8), 43-50.

Aich, S., Josuttes, A., Ovsyannikov, I., Strueby, K., Ahmed, I., Duddu, H. S., Pozniak, C., Shirtliffe, S., & Stavness, I. (2018). DeepWheat: Estimating phenotypic traits from crop images with deep learning. Computers and Electronics in Agriculture, 155, 271–282. https://doi.org/10.xxxx/placeholder

Khan, R., Khan, A., Muhammad, I., & Khan, F. (2025). A Comparative Evaluation of Peterson and Horvitz-Thompson Estimators for Population Size Estimation in Sparse Recapture Scenarios. Journal of Asian Development Studies, 14(2), 1518-1527.

Khan, R., Khan, A., Muhammad, I., & Khan, F. (2025). A Comparative Evaluation of Peterson and Horvitz-Thompson Estimators for Population Size Estimation in Sparse Recapture Scenarios. *Journal of Asian Development Studies*, *14*(2), 1518-1527.

Ahmad, M., Khan, S., Ahmad, R. W., & Rehman, A. A. (2025). COMPARATIVE ANALYSIS OF STATISTICAL AND MACHINE LEARNING MODELS FOR GOLD PRICE PREDICTION. *Journal of Media Horizons*, *6*(4), 50-65.

Ahmad, M., Amin, K., Ali, A., & Ahmad, R. W. (2025). A Comparative Evaluation of Poisson, Negative Binomial, and Zero-Inflated Models for Count Data. *world*, *3*(8).

Ahmad, M., Qamar, H., Rehman, A. A., & Khan, R. (2025). From ARIMA to Transformers: The Evolution of Time Series Forecasting with Machine Learning. Journal of Asian Development Studies, 14(3), 219-233.

Ahmad, M., Rehman, A. A., Khan, R., & Bibi, H. (2025). Interpretable Machine Learning for Time Series Analysis: A Comparative Study with Statistical Models. ACADEMIA International Journal for Social Sciences, 4(3), 4001-4009.

KHAN, R., SHAH, A. M., & KHAN, H. U. (2025). Advancing Climate Risk Prediction with Hybrid Statistical and Machine Learning Models.