IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Corn Leaf Disease Detection Based On Transfer Learning Of Resnet34

[1]Md.Afreen Fathima, [2]U.Kavitha , [3]P.Jithendra Kumar, [4]V K Srividya M [5]Dr.G.Reshma [1-5]Department of Information Technology

[1-5]Prasad V Potluri Siddhartha Institute of Technology, Vijayawada, India

Abstract: Corn is often known as the queen of cereals globally because of its highest genetic yield potential among cereals. It is crucial to promptly and efficiently identify diseases that affect corn leaves to preserve the yield of corn and ensure it is accessible for both humans and animal livestock. A deep learning approach can help enhance this process and improve the accuracy of disease detection. This article leverages the transfer learning Resnet-34 model, aiming to increase the effectiveness of the detection process. The dataset contains images of healthy corn leaves in addition to five diseases that affect the corn leaves. A transfer learning Resnet-34 is pretrained and fine-tuned to increase the effectiveness of image classification. For quantitative analysis, the key statistic is accuracy, with precision, recall, and f1 score serving as supplementing metrics to analyze the model's performance for individual classes. The accuracy of the model obtained is 96% in detecting the disease's effect on corn leaves. Furthermore, precision, recall, and f1 scores for all classes are above 0.90. The transfer learning Resnet-34 model is effective in detecting diseases in corn leaves, as demonstrated by these findings.

***Index Terms*— Convolutional Neural Network (CNN),Transfer Learning, Residual Network, Disease Classification, Preprocessing, Confusion Matrix, Early Stopping, Neural Networks, Validation, Model Performance.**

I. INTRODUCTION

Corn is one of the most widely cultivated crops in the world, grown on more than 118 million hectares and producing around 600 million metric tons annually. Beyond being a food staple, corn is also a valuable resource for producing ethanol, a type of biofuel, while its stalks are used to make plastics and fabrics. However, corn production faces several challenges, as various diseases can negatively impact crop growth, quality, and ultimately, prices. Plants affected by these diseases often exhibit disrupted growth patterns, unusual tissue behavior, and visible signs like yellowing or drying of the leaves.

One such disease, *Puccinia sorghi*, causes common rust in maize, which is particularly damaging to popcorn, sweet corn, and seed corn production. The fungus forms elongated, brown to reddish-brown pustules on both the upper and lower leaf surfaces, which can significantly impact the plant's health. Another pest, the Fall Armyworm (*Spodoptera frugiperda*), is notorious for damaging maize leaves and is challenging to control, but effective management is still possible.

Gray Leaf Spot (GLS) is another fungal disease caused by *Cercospora zeae-maydis*. It starts as small necrotic spots surrounded by yellow halos that are easily seen when the leaves are backlit. These lesions grow into long, circular spots, turning dark gray or tan as they mature. Under moist conditions, dark spores are produced, giving the lesions a "dirty" appearance. Northern Corn Leaf Blight (NCLB), caused by

Setosphaeria turcica, primarily affects seed-producing corn varieties. NCLB lesions appear as grayish-tan spots surrounded by dark borders, often on the lower edges of the leaves, and can also appear on the leaf sheaths and husks.

To combat these challenges, modern technology has proven invaluable, especially in the form of Convolutional Neural Networks (CNNs). CNNs have become an essential tool in classifying images of corn leaves to detect diseases. In this research, we used a transfer learning approach with the ResNet-34 architecture, which helped extract high-quality features for improved disease detection.

To make this technology more accessible to farmers and researchers, we developed a user-friendly, web-based interface using Gradio. This platform allows users to upload images of corn leaves and receive real-time predictions about the disease type, providing valuable insights for better crop management.

Several key aspects were incorporated into this project to ensure accuracy and usability:

1. Various image transformations were applied, such as resizing, random horizontal and vertical flips, Gaussian blur, and normalization, to enhance the model's robustness.
2. We collected data on five different corn diseases to improve the model's ability to detect a wide range of issues, ensuring that it could identify diseases affecting corn reliably.
3. Early stopping techniques were used during training to prevent overfitting, with a patience hyperparameter employed to halt the process once accuracy improvements plateaued.

II. LITERATURE REVIEW

A group of researchers has successfully developed and deployed a prototype for detecting corn leaf diseases using a Convolutional Neural Network (CNN) combined with OpenMP. The system performed exceptionally well, accurately identifying various diseases on the Raspberry Pi platform. In the testing phase, it achieved impressive accuracy rates: 93% for leaf blight, 89% for leaf rust, and 89% for leaf spot detection.

In another line of research, the team explored transfer learning by using Inceptionv3 and Inceptionv4 models with a dataset containing eight categories and 3,503 images. The accuracy of disease detection varied depending on how the models were trained. When all layers were trained, Inceptionv3 achieved 82.8% accuracy, while Inceptionv4 achieved 81% accuracy. When only the final layer was trained, the accuracy decreased slightly, with Inceptionv3 reaching 77.6% and Inceptionv4 at 77%.

Additionally, the researchers used Neuroph Studio as an Integrated Development Environment (IDE) to build a more efficient deep CNN. They embedded convolution and pooling feature extractions into the Neuroph library, utilizing datasets from Plant Village’s online resources. This approach resulted in an impressive overall accuracy of 92.85%, highlighting the framework's practicality.

Another study focused on how different parameters and models affect corn disease diagnosis. The research demonstrated excellent accuracy, particularly with 32-means samples. Diagnostic recalls for diseases like leaf spot, rust, and grey spot reached 89.24%, 100%, and 90.95%, respectively. Different models, including VGG- 16, ResNet-18, Inception v3, and VGG-19, showed varying accuracies depending on the sample size.

Lastly, a research initiative introduced an improved Gradient Boosting Decision Trees (GBDT) technique to enhance the disease detection process. This method was applied to a dataset containing NLB-infected and uninfected leaves, yielding significant improvements in training efficiency and reduced memory usage. The GBDT method achieved an average recognition accuracy of over 92% in just 15 minutes. This innovation is particularly beneficial for smaller, imbalanced datasets, where efficient processing and accurate detection are crucial.

These advancements showcase the ongoing evolution in using artificial intelligence to tackle agricultural challenges, making it easier for farmers to detect and manage corn diseases with higher accuracy and efficiency.

III. DATASET

The Corn Leaf Disease Dataset is a collection of images that feature corn leaves affected by various diseases,It includes a collection of images depicting corn leaves, both healthy and affected by different types of diseases. The dataset provides valuable resources for training machine learning models to identify and classify diseases that commonly affect corn plants.

3.1 Dataset Overview

The dataset includes photos of corn leaves in various states – some are healthy, while others show signs of disease. The images are typically in .jpg or .png formats, which are easy to use for training machine learning models.

There are more than 4,000 images in total, covering both healthy leaves and those with various diseases. This large collection provides enough data for machine learning models to be trained effectively and be able to recognize different diseases accurately.

3.2 Classes and Categories

The dataset consists of images that are labeled based on the type of disease affecting the corn leaves or if the leaves are healthy. The categ ories include:
Common Rust (Label 0) – 1306 images
These images depict corn leaves affected by the common rust disease caused by the Puccinia sorghi fungus. The disease leads to rust-colored pustules on the surface of the leaves.

Gray Leaf Spot (Label 1) – 574 images
This category includes images of leaves infected by Cercospora zeae-maydis, which causes gray leaf spot. The disease is characterized by small necrotic spots surrounded by yellow halos, often visible when leaves are backlit.

Blight (Label 2) – 1146 images
This section contains images of corn leaves affected by blight, a disease that causes rapid deterioration of leaf tissue. The affected leaves show signs of wilting, yellowing, and decay.

Healthy (Label 3) – 1162 images
These are images of healthy, disease-free corn leaves. These images serve as the reference for distinguishing between diseased and healthy leaves in machine learning models.

Common Rust

Blight

Grey Leaf Spot

Healthy

fig. 1. Sample Images from theCorn Leaf Disease Dataset

3.3 Dataset Splitting

We segregated the dataset into three parts to facilitate the training, validation, and testing process. Specifically, 70% of the data, which is 2,195 images, was reserved for training, 20% (627 images) for validation, and the remaining 10% (313 images) for testing.

Classes	Types of Classes	Training Set Images	Validation Set Images	Testing Set Images
0	Blight	802	229	115
1	Common_Rust	914	261	131
2	Gray_Leaf_Spot	401	114	59
3	Healthy	813	232	117

Table1: Dataset Distribution

3.4 Data Preprocessing

We have used PyTorch to preprocess images by setting up three distinct pipelines: train_transform, val_transform, and test_transform. Each pipeline is tailored to specific stages in a computer vision workflow. The train_transform pipeline focuses on augmenting training data. It resizes images to 256x256 pixels and introduces various preprocessing techniques. This process helps prevent overfitting and converts images to PyTorch tensors for compatibility with neural networks. Additionally, normalization standardizes data by adjusting channel means to [0.485, 0.456, 0.406] and standard deviations to [0.229, 0.224, 0.225].

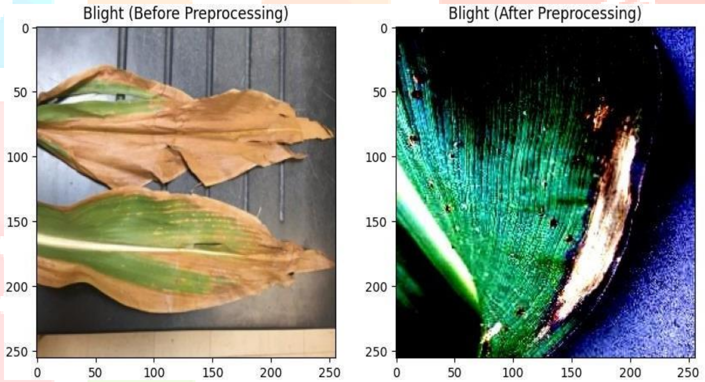


Fig.2. Image of leaf blight before and after preprocessing

IV. PROPOSED SYSTEM

This suggested model introduces convolutional neural network (CNN) models designed to accurately identify various types of corn leaf diseases from images, providing practical utility for agricultural practitioners. The experimentation involved utilizing the Corn Leaf Disease dataset sourced from agricultural repositories. The CNN model proposed in this study is RESNET-34, a pre-trained architecture with 34 layers, tailored for the detection of corn leaf diseases from image data. Fig.2. Image of leaf blight before and after preprocessing

V. METHODS

A. TRANSFER LEARNING

Transfer Learning involves transferring knowledge acquired from a source task to a target dataset. Its objective is to maintain the knowledge gained by crucial neurons, allowing them to retain their ability to extract features from the original domain, thereby enhancing the network's performance on the new dataset

[16]. It's important to note that directly using labeled data from the source domain on a different target domain often leads to subpar performance due to a semantic gap between the two domains, despite representing the same objects. This gap can stem from various factors like different acquisition conditions (such as illumination or view angle) or the use of distinct cameras or sensors. To address this issue, transfer learning methods are introduced, aiming to bridge the semantic gap [17]. Traditionally, these methods involve adopting linear or nonlinear transformations, often through kernel functions, to learn a shared subspace that helps bridge this gap. Transfer learning brings several advantages, notably reducing training time, enhancing neural network performance in most scenarios, and circumventing the need for an extensive amount of data [18].

B. RESNET – 34 ARCHITECTURE

C.

Fig.3. depicts the architecture of resnet-34. It has basic residual blocks which makes training extremely deep neural networks easier. The architecture is as follows:

1. *Input layer*: An RGB image of size 256x256 was passed to the network as input.

2. *First convolution layer*:

-> This layer uses 7x7 kernel size in step 2 and produces 64 output channels.
-> The output was normalized using a batch normalization layer.
-> Relu Introduced nonlinearity using activation function.
-> Max pooling was applied with kernel size 3x3, stride and padding of 2 and 1 respectively.

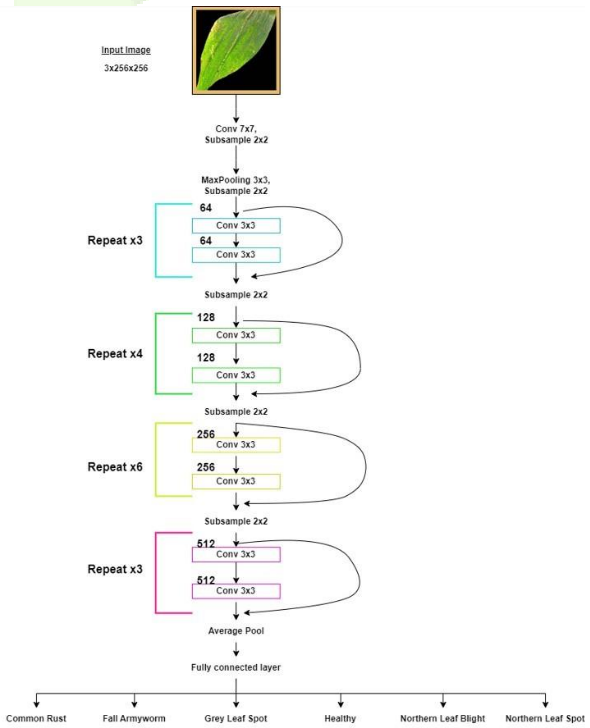
3. *Residual blocks*:

-> Four residual blocks were present in the architecture.
-> Each residual block contains two or more basic blocks.
-> Two convolutional layers with output channels (64,128,256,512) and a batch normalization layer are present in each basic block, along with an activation function.
-> The first residual block consists of three basic blocks.
-> The second residual block has four basic blocks
-> The third residual block has six basic blocks.
-> The fourth residual block consists of three basic blocks.

4. *Skip connections*: Each basic block has a skip connection that bypasses the convolutional layer to alleviate the vanishing gradient problem.

5. *Adaptive average pooling*: The output of the convolutional layer was subjected to average pooling, and the output size was fixed to 1x1.

6. ***Fully Connected Layer*:** The output of adaptive average pooling is flattened and fed to the fully connected layer. The last fully connected layer has 512 input features, resulting in 6 output features, namely the corn disease class.



VI. RESULTS AND ANALYSIS

In this study, we used a batch size of 32 and the no.of epochs as 20. We have also used the concept of early stopping by assigning the patience value 3 while training of the dataset is happening.

A. Accuracy and loss graphs:

The fig.4 shows a graph of accuracy scores over time. The x-axis represents the iterations performed by the model during the training phase. The Y-axis represents the accuracy score. This is a value between 0 and 1 that represents the percentage of correct predictions of the model. The blue line represents the training accuracy, i.e. the accuracy calculated based on the training data. The orange line represents the validation accuracy, that is, the accuracy calculated based on the validation data. The best training accuracy is achieved in 20 epochs with an accuracy of 95% and the best validation accuracy is 94. 5%.

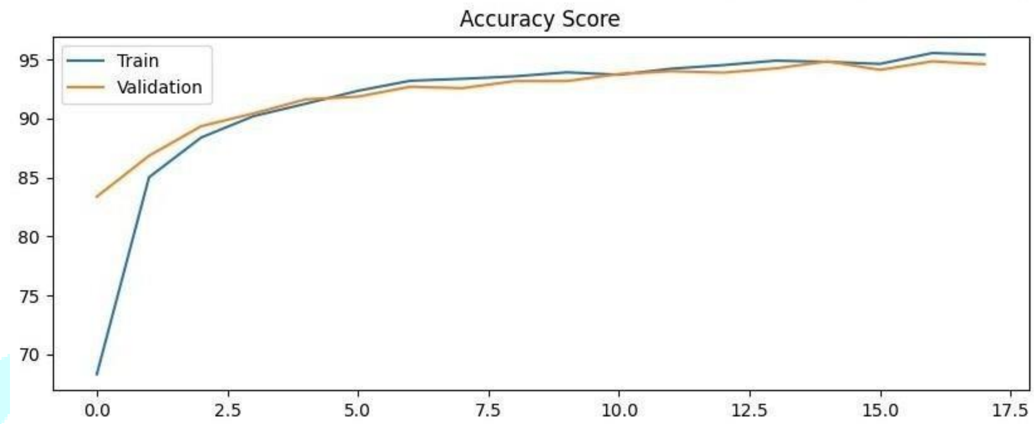


Fig. 4. Graph showing training and validation accuracy

The fig.5 shows a graph of loss over time. The x-axis represents the epochs or iterations that the model passes through during the training process. The y-axis represents the loss or error value of the model prediction at a particular epoch. The blue line represents the training loss calculated from the training data. The orange line represents the validation loss calculated from the validation data. The lowest training loss occurred at epoch 17 and was 0.15%, and the validation loss was 0. 16%.

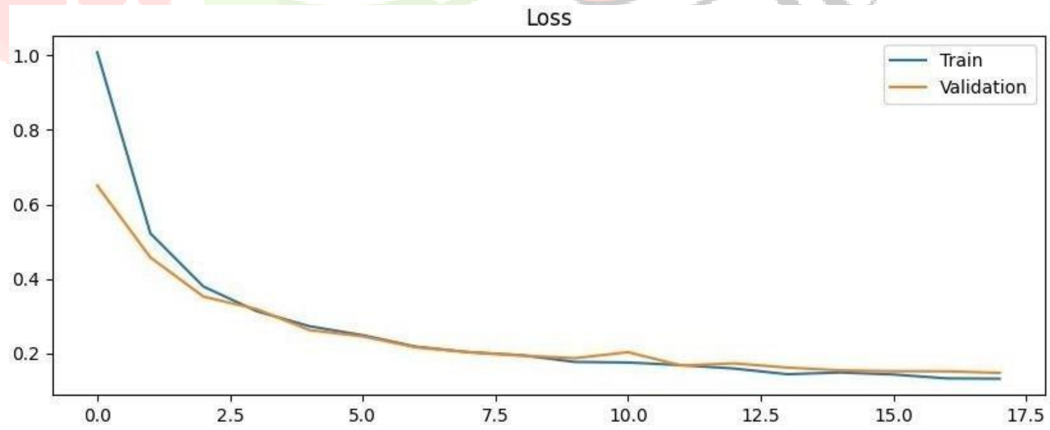


Fig. 5. Graph showing training and validation loss

B. Analysis based on the confusion matrix's configuration:

Confusion matrix:

A confusion matrix, also known as an error matrix or contingency table, is a fundamental tool for evaluating the performance of classification algorithms.Heatmaps are created using sklearn.metrics confusion_matrix() function and Seaborn's heatmap() function. This heatmap represents the model's performance on the test data. Fig. 6. represents a six-class confusion matrix. Out of 58 test images, the model correctly identified

57 as common rust. Out of 44 test images, 39 were recognized correctly and the remaining 5 were recognized as normal. Out of his 52 images in gray leaf spot, 51 were recognized correctly and 1 was recognized as northern leaf spot. Out of 56 test images of healthy images, 55 were recognized correctly and 1 was recognized as general rust. Of his 51 NLB images, 50 were recognized correctly and 1 was recognized as GLS. Of the 56 images in northern leaf spot, 53 were correctly recognized and 3 images were recognized as gray leaf spot, northern leaf blight.

Fig. 6. Confusion matrix for six different classes

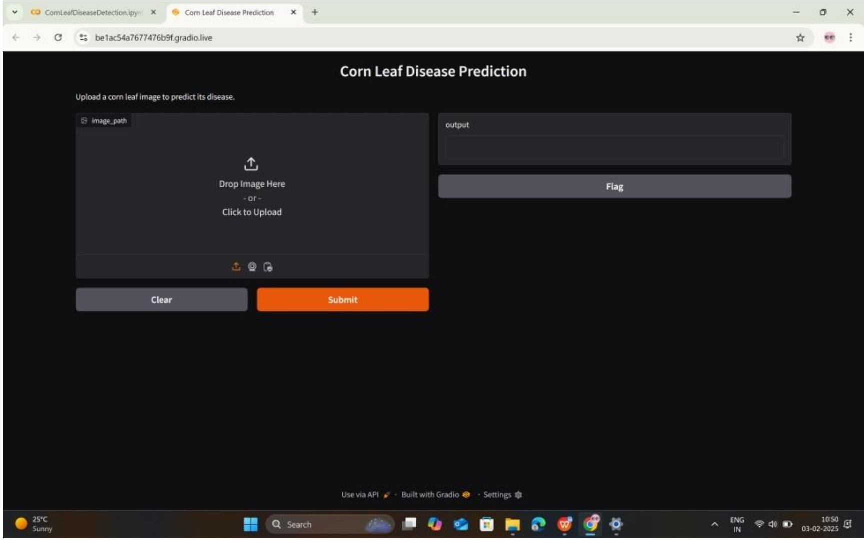
C. *Classification Report:*

Accuracy: Accuracy is the frequency with which the classifier predicts the future correctly. Calculating accuracy involves dividing the total number of guesses by the number of right forecasts
Accuracy=[TP|+|TN|/n [19]
Precision: Precision measures the percentage of correctly categorized samples or instances among the positive samples. As a result, the precision formula may be expressed numerically as
Precision=TP/(TP+FP) [20]
Recall:Recall gauges how well the model can forecast the positives.
Recall=TP/(TP+FN) [21]
F1 Score: The precision and recall weighted average is known as the F1 score.
F1=2.precision.recall/(precision+recall) [22]
Macro Average: This metric calculates the average value of the metric calculated for each class.
Weighted average: Metrics are averaged per class, but the average is weighted by the number of instances in each class. Support represents the number of images in the test data set.
Support: The number of images in test set of each class.

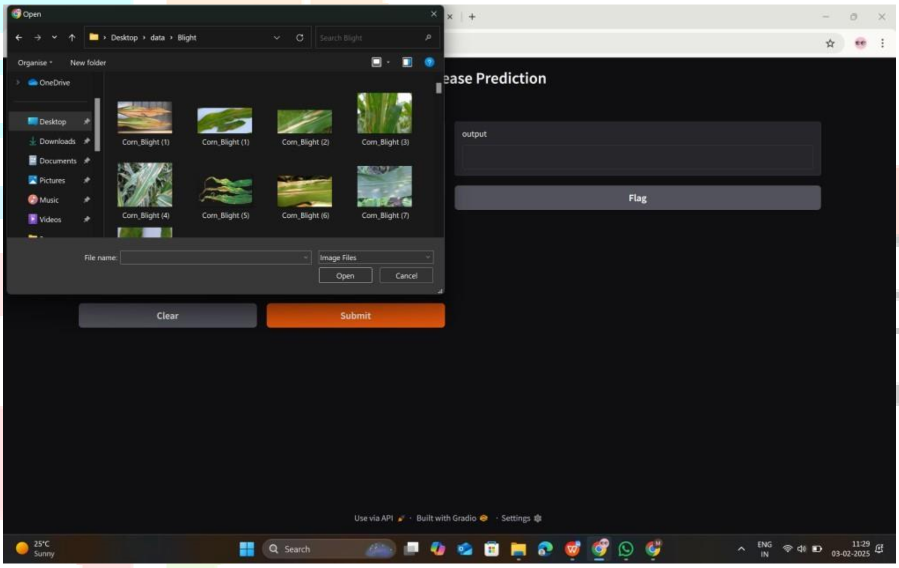
	Precision	Recall	f1-score	Support
0	0.91	0.88	0.89	115
1	0.98	0.96	0.97	131
2	0.80	0.88	0.84	59
3	1.00	1.00	1.00	117
accuracy			**0.94**	**422**
macro avg	0.92	0.93	0.93	422
weighted avg	0.94	0.94	0.94	422

Table 2. classification metrics of all classes along with macro average and weighted average

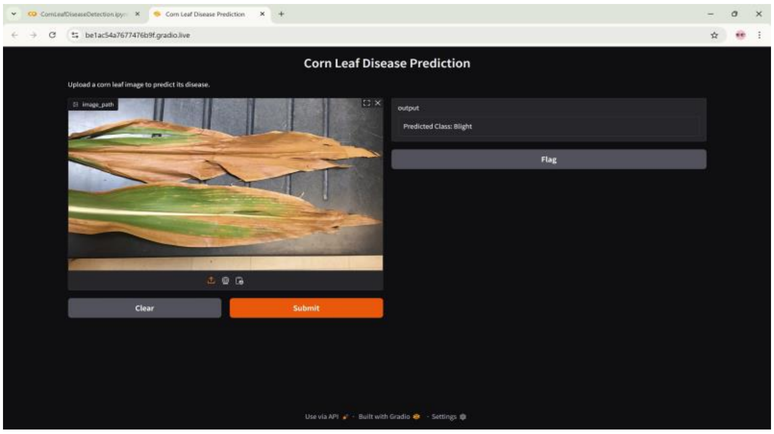
VII. OUTPUT SCREEN



- When the application is started,the user is presented with the main interface,which contains a title, file upload option and submit button.
- The title displayed at the top reads “”Corn Leaf Disease Detection”.
- Drag-or-drop file area allows users to select or upload image from their local storage for processing.



- The application supports common image formats such as JPEG,JPG and PNG.
- Once the user selects an image,click on submit button to predict the disease type.
- The user can review the prediction and flag it if they disagree with the result.
- The user can choose to clear the image or submit another image for prediction.



- User uploaded corn leaf image image uploaded shows a corn leaf that is discolored and has brown spots, so model predicted that the leaf is infected with Blight.

VIII. CONCLUSION

The proposed system for identifying corn leaf diseases using deep learning has demonstrated exceptional effectiveness. By utilizing the pre-trained ResNet-34 model and transfer learning, the system achieved high accuracy in detecting five distinct corn leaf conditions, including healthy samples. The dataset consisted of 4,188 images, divided into training, validation, and testing sets in a 70:20:10 ratio, and training was optimized using early stopping to minimize loss and enhance performance. The model achieved a remarkable accuracy of 9%, with precision, recall, and F1 scores exceeding 0.90 for all classes. The convolutional layers and residual blocks in ResNet-34 played a crucial role in accurately extracting features and identifying diseases. These results highlight the potential of deep learning approaches for real-time crop disease management In addition to the model, a user-friendly Gradio interface was developed, allowing users to upload corn leaf images and receive predictions about their disease type (e.g., Blight, Healthy, Common Rust, Grey Leaf Spot). The interface includes essential functionalities like image upload, result display, input clearing, and submission flagging, ensuring an interactive and accessible experience. It operates on a Colab backend with GPU support, enabling fast processing and deployment. Future work could focus on extending the model's capabilities to detect pathogens and predict disease severity, further aiding farmers and researchers in proactive crop health management. The integration of an accessible frontend ensures this solution is practical and impactful for real-world applications.

IX. REFERENCES

[1] Z. Jabran, "Maize: cereal with a variety of uses," DAWN–Business, 2007. [Online]. Available: http://wwwdawn.com/2007/03/12/ebr5.htm.

[2] A. K. S. H. D. Kshyanaprava Panda Panigrahi, "A CNN Approach for Corn Leaves Disease Detection to support Digital Agricultural System," in *IEEE*, 2020.

[3] K. Wise, "Common and Southern Rusts".

[4] M. K. T. K. Srisha Bista, "Fall armyworm: Menace to Nepalese farming and the integrated management approaches," *International Journal of Environment, Agriculture and Biotechnology*, pp. 1011-1018, 2020.

[5] M. Henrickson, "Gray Leaf Spot of Corn," Pioneer Agronomy Sciences, 20 August 2019. [Online]. Available: https://www.pioneer.com/us/agronomy/gray_leaf_spot_cropfocus.

[6] A. R. D. Mueller, "Northern Corn Leaf Blight of Corn," Crop Protection Network, 3 March 2019. [Online]. Available: https://cropprotectionnetwork.org/encyclopedia/northern-cornleaf-blight-of-corn.

[7] D. Malvick, "Northern corn leaf spot," University of Minnesota Extension, 2018. [Online]. Available: https://extension.umn.edu/corn-pest-management/northern-cornleaf-spot.

[8]V. L. D. H. F. P. R. S. Y. P. Tegar Arifin Prasetyo, "Corn Plant Disease Classification Based on Leaf using Residual Networks-9 Architecture," *IJECE*, vol. 13, pp. 2908-2920, 2023.

[9] D. Q. H. M. C. Mingyu Gao, "A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects," *Forests*, vol. 12, no. 2, 2021.

[10] R. A. I. P. T. D. G. Dionis A. Padilla, "Detection of Corn Leaf Diseases Using Convolutional Neural Network With OpenMP Implementation," in *12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2020.

[11]J. W. Xiaolin Sun1, "Identification of maize disease based on transfer learning," *Journal of Physics: Conference Series*, vol. 1437, no. 1, 2020.

[12] M. S. Malusi Sibiya, "A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy," *AgriEngineering*, no. 1, pp. 119-131, 2019.

[13] J. L. C. A. H. Helong Yu, "Corn Leaf Diseases Diagnosis Based on KMeans Clustering and Deep Learning," *IEEE Access*, pp. 43824- 143835, 2021.

[14] T. H. L. a. Y. C. Xiao, "Corn disease identification based on improved GBDT method," in *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, 2019.

[15] A. S. D. G. A. E. J. G. Ahmad, "Cd&s dataset: Handheld imagery dataset acquired under field conditions for corn disease identification and severity estimation," 2021.

[16] X. G. Z. L. D.-Z. X. Kafeng Wang, "Pay Attention to Features,Transfer Learn Faster CNNs," in *ICLR*, 2020.

[17] L. Z. Q. W. W. Jianzhe Lin, "DT-LET: Deep transfer learning by exploring where to transfer," *Science Direct*, vol. 390, pp. 99-107, 2020.

[18] P. Sharma, "Understanding Transfer Learning for Deep Learning," AnalyticsVidhya, 7 December 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/understandingtransfer-learning-for-deep-learning/.

[19]G. S. NagaSubrahmanyeswari.N, "Classification Evaluation Metrics in Machine Learning," *Journal of Emerging Technologies and Innovative Research(JETIR)*, vol. 9, no. 10, 2022.

[20] K. P. Shung, "Accuracy, Precision, Recall or F1?," Towards Data Science, 15 March 2018. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1- 331fb37c5cb9.

[21] rashida, "Learn Precision, Recall, and F1 Score of Multiclass Classification in Depth," regenerative, 10 October 2022. [Online]. Available: https://regenerativetoday.com/learn-precision-recalland-f1-score-of-multiclass-classification-in-depth/.

[22] S. Goyal, "Evaluation Metrics for Classification Models," AnalyticsVidhya, 20 July 2021. [Online]. Available: https://medium.com/analytics-vidhya/evaluation-metrics-forclassification-models-e2f0d8009d69.