ISSN No:-2456-2165

Corn Leaf Disease Detection
(The Crop Master)

Aditi Maurya, Varada Nakhate, Ananya Maurya, Nuha Modak
Computer Engineering
SIES Graduate School of Technology, Nerul, Navi Mumbai

Abstract:- Corn production is a vital component of the agricultural industry, serving critical roles in areas such as biofuel production and the global food supply chain. Moreover, it supports household industries through small-scale cultivation. However, corn crops face significant risks, including susceptibility to diseases that can severely impact agricultural yields. Furthermore, extreme weather events like cyclones and unpredictable temperature fluctuations can aggravate the spread of these diseases. Given the limitations of the human eye in detecting leaf sickness or disease, there is a pressing need for a rapid and intelligent disease detection process, utilizing advanced deep learning techniques. To address this challenge and enhance crop yield, recent advancements in smart devices have enabled the implementation of Convolutional Neural Network (CNN) models for the training and testing of corn leaf images. This innovative approach offers a time-efficient solution for the early detection of leaf diseases, ultimately strengthening the nation's support for digital agriculture.

***Keywords**:- Convolutional Neural Network; detection; digital agriculture; leaf sickness; images*

I. INTRODUCTION

Plant diseases pose a significant threat to agricultural production, adversely affecting plants through fungal, viral, and bacterial infections that can target leaves, branches, and fruits. These diseases manifest in various forms, such as alterations in color, shape, or edges. Detecting these ailments often requires the expertise of a plant pathologist, which can be both costly and not always immediately evident. Corn, as a globally prominent agricultural crop, holds a pivotal role in diverse industries, including cooking oil, starch, flour, sugar, biofuel, and more. To address the challenges of disease detection and optimize crop production, machine learning algorithms have been deployed for predictive purposes, yielding accurate results over extended periods. Experts harness the power of these algorithms to discern subtle changes in leaf color and other vital indicators. In developing countries like India, the economy relies heavily on agriculture, making it a cornerstone of livelihoods. However, the agricultural sector faces significant challenges due to plant diseases, which not only diminish the quality but also reduce the quantity of agricultural products. Compounding the issue is the fact that certain plant diseases remain hidden during their early stages, only becoming visible in the advanced stages. The early diagnosis of plant diseases is of paramount importance as it enables timely intervention and effective disease control. This diagnostic process often involves experts who possess the unique ability to discern subtle changes in leaf color, texture, and other vital indicators. However, it's worth noting that even among these experts, there can be variability in disease identification, leading to instances where different experts label the same disease differently. Corn diseases can manifest in various parts of the plant, including the leaves, stems, rhizomes, or even throughout the entire plant. Among these, corn leaves often display a diverse range of symptoms, with some of the most prevalent infections being gray leaf spot, common rust maize disease, northern leaf blight, and brown spot. The proposed approach utilizes Convolutional Neural Networks (CNNs) to address the challenge of identifying and classifying various corn leaf conditions. In our dataset, comprising 3997 images of corn leaves, we distinguish four distinct categories: common rust, grey leaf spot, blight leaf, healthy leaf images, and even background images devoid of leaves. Our system relies on these images to capture and meticulously analyze the leaves, discerning whether they are infected or healthy. The workflow begins with essential image pre-processing techniques, including segmentation and grayscale conversions. Subsequently, we embark on the training and detection phase using a deep learning algorithm, specifically a Convolutional Neural Network (CNN). During training, we primarily focus on a dataset containing infected leaf images and normal leaf images. The significance of safeguarding crops against diseases cannot be overstated in the context of global food security. Early-stage disease recognition plays a pivotal role in this protection. Traditionally, the identification and detection of crop leaf diseases have relied on the expertise of agricultural technicians. However, our proposed method leverages computer vision technology to automate this process, eliminating subjective judgment errors and achieving exceptional efficiency. This advancement is paramount for enhancing crop cultivation quality and securing our global food supply. Agricultural productivity stands as a pivotal driver of economic growth, and this underscores the critical importance of plant disease detection within the agriculture sector. Plant diseases are a prevalent challenge, and their detection and management are paramount. Failing to take necessary precautions in this domain can lead to severe repercussions, with plants bearing the brunt of the consequences. Such repercussions manifest in the form of diminished quality, reduced quantity, and overall decreased productivity of the respective agricultural products. As such, effective plant disease detection measures are essential to safeguard both crop health and economic prosperity in this vital sector. For instance, the United States has pine trees that are susceptible to a dangerous illness called small leaf disease. The use of an automatic method for plant disease detection is advantageous because it lessens the amount of labor required to monitor large crop farms and can identify disease symptoms at their earliest stage, when they first

ISSN No:-2456-2165

emerge on plant leaves. Pest infestations have a significant impact on plant productivity, so ongoing surveillance is necessary to prevent this issue. The suggested approach employs photographs to take a leaf and determine whether it is infected. Convolution neural network, a deep learning algorithm, is employed for analysis. The dataset with photos of diseased and healthy leaves is considered for training. This entails segmentation and grayscale conversions as part of the image pre-processing, followed by training and detection. In the suggested approach, plant leaves are divided into normal and infected categories.

II. PROPOSED SYSTEM

The significance of disease identification in plants cannot be overstated within the realm of agriculture. As plant diseases are widespread, their timely detection is crucial for the agricultural sector. Neglecting necessary precautions in this domain can result in severe consequences, directly affecting the quality, quantity, and overall productivity of agricultural products. Plant diseases stand as a major contributor to agricultural production losses, with bacteria, viruses, and fungi being the culprits behind these detrimental infections. These pathogens can wreak havoc on various parts of a plant, including leaves, branches, and fruits, causing a spectrum of changes in color, shape, or edges. Detecting such diseases often necessitates the expertise of a plant pathologist, a process that can be both expensive and challenging due to the sometimes-subtle nature of these infections. In recent years, the rapid advancement of deep learning and Convolutional Neural Networks (CNNs) has revolutionized image recognition. In our specific case, we aim to accurately classify maize leaves from a dataset, considering their health condition, whether they are affected by common rust, grey leaf spot, blight, or remain healthy. To achieve this, we employ the CNN approach, capitalizing on a dataset comprising 3997 maize leaf images that are categorized into these four conditions, along with background images lacking leaves.

Our innovative approach relies on these images to effectively analyze and determine whether a leaf is infected or not. This analysis is made possible by utilizing the power of Convolutional Neural Networks, a deep learning algorithm. The training process incorporates image preprocessing steps, including segmentation and grayscale conversion, followed by the training and detection phases. This comprehensive strategy is poised to play a pivotal role in enhancing disease management and ultimately securing agricultural productivity.

III. ARCHITECTURE AND FRAMEWORK

The dataset provided is specifically designed for the classification of corn or maize plant leaf diseases and comprises a comprehensive collection of images. It includes the following disease category:

- **Common Rust**: The dataset contains a substantial collection of 954 images related to common rust affecting corn or maize plants.
- **Background Without Leaves:** This category encompasses images where the leaf is not accurately identified or is not the primary subject of the image.
- **Cercosporin Leaf Spot:** Within this category, you'll find images depicting corn or maize leaves featuring characteristic spots caused by Cercosporin leaf spot disease.
- **Common Rust:** Images in the "Common Rust" category showcase corn or maize leaves displaying a rust-like surface, indicating the presence of common rust disease.
- **Northern Leaf Blight:** This category comprises images of corn or maize leaves afflicted by Northern Leaf Blight, a disease primarily affecting the foliar part of the leaf.
- **Healthy Leaf:** In this category, you will discover images of perfectly healthy corn or maize leaves that exhibit no discernible defects or signs of disease.

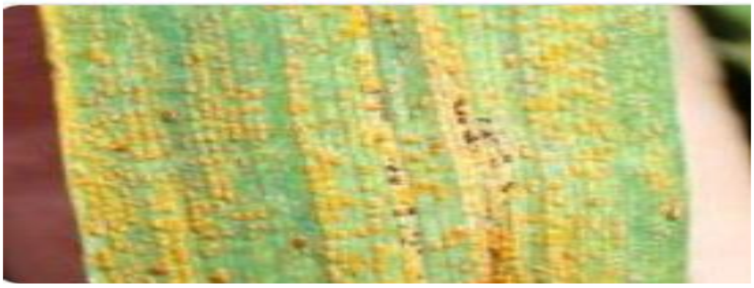


Fig. 1: Gray Leaf Spot Images- 410 Images

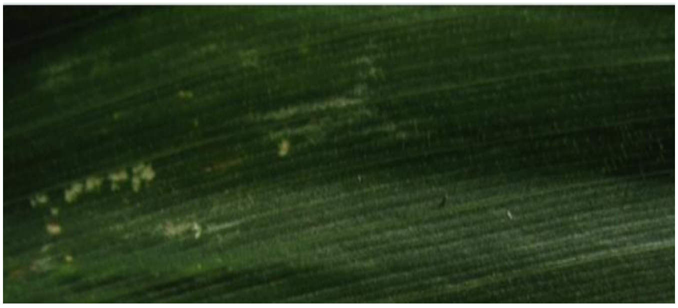


Fig. 2: Blight Leaf -788 Images

ISSN No:-2456-2165


Fig. 3: Healthy Leaf Images - 930 Images


Fig. 4: Background Without Leaves - 915 Images

IV. ALGORITHMS AND PROCESS DESIGN

A Convolutional Neural Network (CNN) is a specialized deep learning neural network designed to effectively process structured arrays of data, particularly well-suited for tasks involving images. One of the remarkable capabilities of CNNs lies in their adeptness at detecting intricate design elements within input images. This includes the identification of features such as lines, gradients, circles, or even complex structures like eyes and faces. This inherent ability makes convolutional neural networks exceptionally reliable and robust for tasks related to computer vision. A notable advantage of CNNs is their ability to work directly on input images without the need for extensive pre-processing. This means that CNNs can efficiently handle underexposed or minimally processed images, further enhancing their utility in various image recognition and analysis tasks.

- **Kaggle:** Kaggle is a globally recognized crowdsourcing platform that attracts data scientists from all corners of the world. It serves as a training ground and a source of challenging tasks for professionals and enthusiasts alike, focusing on various domains within the realms of data science, machine learning, and predictive analytics.
- **Python IDLE:** Python IDLE is a dedicated integrated development environment (IDE) tailored specifically for Python programming. This user-friendly environment is designed to facilitate the creation, editing, and execution of Python 2.x or Python 3 programs. Notably, Python IDLE seamlessly integrates with the Python interpreter, providing a graphical user interface (GUI) that enhances the Python development experience.
- **Concepts:** Confusion Matrix: In the context of binary classification problems, a classifier or classification model generates a tabular representation detailing the count of correct and incorrect predictions, or the alignment between actual and predicted values. This performance measurement tool, known as a confusion matrix, plays a pivotal role in evaluating the effectiveness of machine learning algorithms.
- **OpenCV:** OpenCV, short for Open-Source Computer Vision, is a comprehensive framework designed to standardize operations in the field of computer vision. It provides a versatile platform for performing various computer vision tasks and seamlessly integrates system behavior into financial products, offering a robust and standardized environment for vision-based operations.

V. EXPERIMENTS AND RESULTS

2	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
3	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
4	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
...	...	...
1653	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
1654	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
1655	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy
1656	/kaggle/input/corn-or-maize-leaf-disease-datas...	Blight
1657	/kaggle/input/corn-or-maize-leaf-disease-datas...	Healthy

1658 rows × 2 columns

A. Dataset

The dataset features a diverse collection of images showcasing various types of leaves and the associated leaf faults. Broadly, these images have been classified into four distinct groups: Blight, Common Rust, Grey Leaf Spot, and Healthy. To elaborate, "Blight" refers to a category encompassing several plant diseases characterized by abrupt and severe symptoms such as yellowing, browning, spotting, wilting, or even the complete demise of leaves, flowers, fruit, stems, or the entire plant. "Common Rust," also known as "Brown Rust," is named after the brown coloration of the circular reinspires found on the leaf surfaces. This type of rust disease is identifiable by these distinctive circular brown structures. "Yellow Rust," alternatively referred to as "Stripe Rust," is identifiable by the presence of yellow stripes on the leaf surfaces. In its early stages, these lesions may appear tan or brown before the onset of fungal sporulation. Lastly, "Healthy" leaf images depict leaves that are entirely free from any discernible defects or signs of disease. These categorizations encompass a wide range of leaf conditions and faults, facilitating the development and evaluation of machine learning models for precise leaf disease classification and the identification of healthy leaves.

```
)

Found 560 validated image filenames belonging to 4 classes.
Found 140 validated image filenames belonging to 4 classes.
Found 300 validated image filenames belonging to 4 classes.

[18]: train_data.next()[1]

[18]: array([1., 2., 3., 3., 3., 3., 3., 0., 0., 3.], dtype=float32)

[19]: import matplotlib.pyplot as plt
      from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPool2D
      from tensorflow.keras.preprocessing.image import ImageDataGenerator
      from tensorflow.keras.optimizers import Adam
```

B. Validating Dataset

In our validation process, we initially categorize the images into four distinct classes. Following this categorization, we convert their data type into arrays and specifically set the data type as float. This transformation to arrays with a float data type simplifies further data manipulation and analysis. To support these data processing steps, we begin by importing all the necessary libraries and dependencies. These libraries will provide the essential tools and functions required for efficient handling and analysis of the image data. This combined workflow streamlines our image validation and prepares the data for subsequent analysis and modeling.

```
validation_generator = train_datagen.flow_from_directory(
    root_dir, # same directory as training data
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation')

Found 3352 images belonging to 4 classes.
Found 836 images belonging to 4 classes.

[17]: validation_generator[0][1][1]

[17]: array([0., 1., 0., 0.], dtype=float32)
```

ISSN No:-2456-2165

C. Training and testing the data

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 120, 120, 3)]     0

 conv2d (Conv2D)             (None, 118, 118, 16)      448

 max_pooling2d (MaxPooling2D  (None, 59, 59, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 57, 57, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 28, 28, 32)       0
 2D)

 global_average_pooling2d (G  (None, 32)               0
 lobalAveragePooling2D)

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 5,121
```

D. Table with all layers

In the CNN model, we begin with a structured architecture that consists of several key layers, each playing a unique role in the overall image processing. Here is a breakdown of the basic structure:

- **Input Layer:** Serving as the foundation, the input layer represents the pixel matrix of the image, acting as the entry point for the entire CNN.
- **Convolutional Layer:** This layer is responsible for extracting image features. Low-level convolutional layers capture fundamental features like edges, lines, and corners. These lower-level features are then used as input for higher-level convolutional layers, which learn more abstract and complex features.
- **Max Pooling:** Following individual convolutional layers, max pooling is often applied. Max pooling helps reduce the dimensionality of the images by downscaling the number of pixels in the output from the preceding convolutional layer.
- **Global Average Pooling:** This pooling operation is designed to replace fully connected layers in classical CNNs. It aims to generate one feature map for each category in the classification task, typically in the last layer of the model.
- **Dense Implementation:** The CNN model includes a large dense layer with 512 units, followed by the final layer responsible for computing SoftMax probabilities for each of the 10 categories, corresponding to the 10 digits.

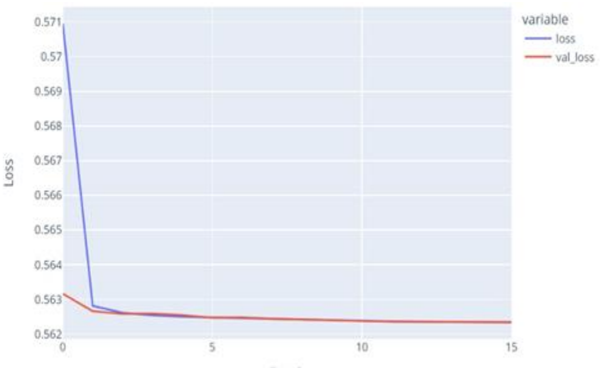
When we print out the model's summary, it provides a structured table with details about each of these layers and their configurations, offering insights into the architecture and parameters of the CNN model.

```
plt.figure(figsize=(6,6))
sns.heatmap(cm,annot=True,fmt='g',vmin=0 ,cmap='Blues',cbar =False)
plt.xticks(ticks=np.arange(4)+0.5,labels =['Blight','Commonrust','Grayleafspot'
plt.yticks(ticks=np.arange(4)+0.5,labels =['Blight','Commonrust','Grayleafspot'
plt.xlabel('predicted')
plt.ylabel('actual')
plt.title('confusion matrix')
plt.show()

print('CLASSIFICATION REPORT',clr)
```

```
75]: evaluate_model(model,test_data)

Test Accuracy: 99.33%
30/30 [==============================] - 2s 78ms/step
/opt/conda/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarn
```

E. Classification Report

In this section, we display the classification report using Matplotlib, showcasing the test accuracy, which stands at an impressive 99.3%.



ISSN No:-2456-2165

F. EEpoch and Loss

During the training process, all the data is utilized exactly once within a single epoch. An epoch constitutes a full pass through the entire dataset. Within each epoch, one or more batches, denoted as "e," are formed, where a subset of the dataset is used to train the neural network. Each iteration in the training process involves going through a batch of training examples. The term "loss" refers to the computed loss value over the training data after the completion of each epoch. This loss value represents what the optimization process seeks to minimize during training. In essence, the objective is to minimize this loss value as much as possible, as a lower loss indicates a better-performing model.

A Confusion Matrix is a valuable tool for assessing the classifier's ability to correctly identify instances belonging to different classes. It is also referred to as a contingency matrix. In the context of the confusion matrix:

Each row signifies an actual class.

Each column represents a predicted class.

Specifically, a 2x2 confusion matrix is denoted as:

		Predicted Class
Actual Class	1	
	0	

- **TP (True Positives):** These are values that are predicted to be true and are actually true.
- **TN (True Negatives):** These represent values that are predicted to be false and are actually false.
- **FN (False Negatives):** FN represents values that are predicted to be false but are, in fact, true.
- **Precision:** Precision is a measure of exactness. It determines what percentage of tuples labeled as positive are actually positive.
- **Precision:** Precision is calculated as TP divided by the sum of TP and FP. Precision = TP / (TP + FP)
- **F-Score:** The F-Score is the harmonic mean of precision and recall, giving equal weight to both. It's also referred to as the F-measure. F-Score = (2 x Precision x Recall) / (Precision + Recall)
- **Support:** Support for itemset A is calculated as the number of transactions in which A appears, divided by the total number of transactions.
- Support(A) = Number of transactions in which A appears / Total number of transactions
- **Confidence:** Confidence for the association rule A -> B is calculated as the support of the combined itemset AUB divided by the support of itemset A.
- Confidence (A -> B) = Support (AUB) / Support(A)

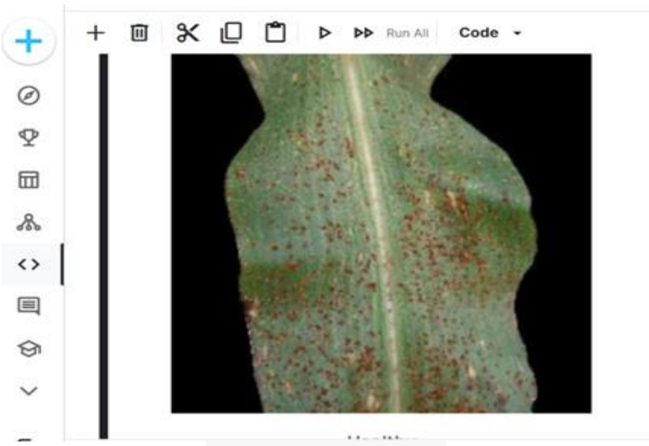


Fig. 5: Testing Images

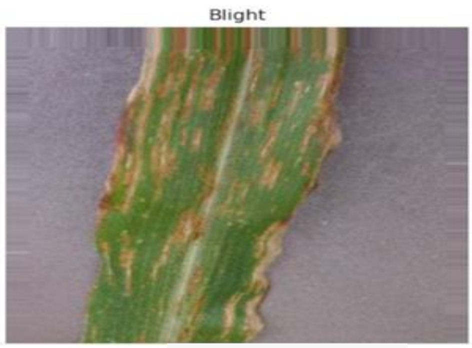


Fig. 6: Testing Images

VI. CONCLUSION

To facilitate automatic illness detection, it is imperative to develop autonomous systems capable of handling extensive data collection efficiently. These systems should be equipped with effective decision-making tools to enable rapid analysis of the risk associated with each leaf disease. This approach empowers benefit managers to take timely and informed actions. Automatic disease detection, in contrast to slower and more subjective traditional human inspection methods, offers the advantage of swift and accurate plant disease analysis. This transition also enhances the safety of the survey process. Non-destructive testing emerges as a particularly successful method for the automatic detection of diseases. The manual inspection method often struggles to objectively evaluate deterioration. Image-based leaf disease detection for non-destructive examination is gaining popularity due to its efficacy. However, several challenges persist in image-based detection, including random shapes and irregular sizes of leaves, unclear images, and various sources of noise such as irregular lighting conditions and shading, as well as blemishes in the acquired images.

REFERENCES

[1.] Patel, Sneha & Jaliya, Dr & Patel, Pranay. (2020). A Survey on Plant Leaf Disease Detection. “International Journal for Modern Trends in Science and Technology.” 6. 129-134. doi: 10.46501/IJMTST.

[2.] Kempegowda, Balakrishna & Rao, Mahesh. (2019). Tomato Plant Leaves Disease Classification Using KNN and PNN. “International Journal of Computer Vision and Image Processing.” 9. 51-63. doi: 10.4018/IJCVIP.2019010104.

[3.] Suresh, Annamalai. (2012). “AN EFFICIENT TEXTURE CLASSIFICATION SYSTEM BASED ON GRAY LEVEL CO-OCCURRENCE MATRIX.” International Journal of Computer Science and Information Technology & Security (IJCSITS), 2, 793-798.

[4.] Harte, Emma. (2020). “Plant Disease Detection using CNN.” doi: 10.13140/RG.2.2.36485.99048.

[5.] Jadhav, Sachin. (2019). “Convolutional Neural Networks for Leaf Image-Based Plant Disease”Classification. IAES International Journal of Artificial Intelligence (IJ-AI). 8. 328. doi: 10.11591/ijai.v8.i4.pp328-341.

[6.] Mohanty, Sharada & Hughes, David & Salathe, Marcel. (2016).” Using Deep Learning for Image-Based Plant Disease Detection.” Frontiers in Plant Science. 7. doi: 10.3389/fpls.2016.01419.