# Customized U-Net CNN Model for Tomato Leaf Based Disease Classification

# LAVANYA K, P. A. MANIAR, T. R. ROKADE, N. JAIN,

# RAMANATHAN LAKSHMANAN

# School of Computer Science and Engineering, Vellore Institute of Technology,632014 Vellore, India

# E-mail: [lavanya.k@vit.ac.in](mailto:lavanya.k@vit.ac.in) , [parshvaanish.maniar2020@vitstudent.ac.in](mailto:parshvaanish.maniar2020@vitstudent.ac.in), [tejasrahul.rokade2020@vitstudent.ac.in](mailto:tejasrahul.rokade2020@vitstudent.ac.in), [naman.jain2020@vitstudent.ac.in](mailto:naman.jain2020@vitstudent.ac.in), lramanathan@vit.ac.in

# Abstract

The pests and agricultural diseases are estimated to be responsible for the loss of ten percent of the annual productivity throughout the globe. Most farmers are unable to identify visually whether a crop has been infected with a disease or not. This is due to a lack of crop morphological knowledge, or a lack of technical knowledge about disease forecast, or temporal features understanding in protecting the crops from being afflicted in the future. The high variability and complexity of leaf diseases, which makes accurate identification difficult, is the principal barrier in tomato leaf disease detection. In our method, customized U-Net CNN (Convolutional Neural Network) associated image processing techniques facilitate the preprocessing of tomato leaf images, enhancement of leaf image quality for feature extraction, and extraction of pertinent information for accurate leaf disease diagnosis. CNN enables efficient feature extraction, whereas U-Net enables accurate segmentation and localization of disease areas in tomato leaf images, resulting in enhanced disease detection and diagnosis. Our model uses images of tomato leaves, to construct a deep learning model that uses a sequential model to identify illnesses that affect plants. After using the U-Net architecture for image segmentation and localization, the convolutional neural network is capable of carrying out classification of tomato leaf diseases witppph an accuracy of 92.77% on training dataset, 94.85% on testing dataset and 83% on random generated dataset.

***Keywords: Deep Learning, Image Processing, GLCM, U-Net CNN, Leaf Disease Detection***

# Introduction

In India, pests and diseases pose a significant hazard to agricultural productivity, resulting in approximately 290 billion rupees in annual losses. Approximately 5,000 of the approximately 30,000 plant diseases identified worldwide are prevalent in India. It is estimated that each year, fungal infections alone reduce agricultural output by approximately 5 million tones. Given that approximately 70% of India's population relies on agriculture for a living, the impact of plant maladies becomes an urgent concern. Despite the fact that producers have access to a variety of crop options and pesticides, plant maladies continue to have a negative impact on the quality and quantity of agricultural products. With a population of 1.40 billion and approximately 95.8 million farmers, India's agricultural sector has the potential to substantially improve the living conditions of farmers, stimulate employment, and promote overall development. Currently, 18% of India's GDP is contributed by agriculture, highlighting the significance of effectively addressing these challenges.

We chose the tomato plant leaf specifically due to its adaptability. This plant is economically significant on a global scale and is extensively cultivated for commercial purposes, making it a practicable and pertinent option for disease detection and segmentation. Additionally, they are susceptible to numerous maladies caused by insects, pathogens, and fungi. This presented us with the challenge of creating a more effective and accurate model for the populace. Tomatoes have leaves that are composed of pinnate or palmate leaflets. By observing leaflet symptoms and patterns, this leaf form facilitates disease identification. Tomato plants may be affected by early, late, bacterial, or septoria leaf spot. The observable symptoms of these leaf diseases make them optimal for leaf disease detection models.

A lot of attempts have been made to detect crop diseases but each had it’s own limitations. [1] The farmers take decisions based on experience which is a very subjective, time-consuming, laborious, and inefficient method. [4] ReLu has a disadvantage that the derivative of the function is zero for negative values and leads to problems like neuronal necrosis.[7] During image acquisition and classification of the images a lot of uneven background was found which made disease identification and classification difficult.[8] The leaves with different diseases are similar to each other which causes trouble in disease detection.[10] Another restriction is the present availability of datasets that do not include photographs collected and annotated from real-world scenarios, and the inability of the suggested approaches to identify numerous illnesses or many instances of the same disease in a single image.[11] C3D-U-Net has its own limitations, including the need for large amounts of annotated CT scans for training and the reliance on well-defined and annotated ground truth data. Moreover, the model's performance may degrade in cases of severe COVID-19 infections or when images have low resolution or high noise.[12] The 3D Multi-scale Multi-attention U-Net for Automated Hippocampal also includes issues like as the necessity for a substantial quantity of annotated MRI data for training, the possibility of overfitting to the training data, and the need for careful hyperparameter tuning. Also, the model's performance may vary with different MRI acquisition protocols and may be sensitive to image noise and variability in image intensity.[13] Deep Learning model-based medical image segmentation also has some similar problems like it is sensitive to imaging variability and needs for careful hyperparameter tuning. These models have limited generalizability across different imaging modalities and acquisition protocols. [14] The U-Net model on the other hand requires a significant number of annotated images of persimmon leaves with disease labels to train effectively, which can be a challenge to obtain.[15] Another common problem faced is the potential for overfitting, where the model may perform well on the training data but poorly on new, unseen data.

In this paper, we propose customized U-Net CNN architecture in which U-Net is renowned for its accuracy in image segmentation, making it an ideal choice for applications requiring precision. We intend to construct an artificially intelligent model capable of identifying and classifying tomato crop diseases by leveraging the capabilities of U-Net for image feature extraction. Our work contributes to the development of reliable and robust systems for early detection and monitoring of tomato leaf diseases through the incorporation of GLCM feature extraction. Advanced imaging techniques and robust deep learning models have the potential to significantly improve agricultural practices and crop yield. This model will enable producers to swiftly determine whether their crops are infected, thereby providing valuable insight into the most appropriate course of action. Upon uploading an image of a diseased tomato leaf, our model will not only detect the presence of disease, but also classify the disease type.

# Literature Survey

Khirade and Patil [1] proposed a study on plant disease identification using leaf images, discussing segmentation and feature extraction techniques for efficient agricultural production. [2] Sardogan and Tuncer used CNN to effectively identify tomato leaf diseases, leveraging deep learning's automatic feature extraction and achieving accurate classification. [3] Shrestha and Deepsikha developed a CNN-based method for plant disease identification, achieving 88.80% accuracy on test data. Their research can aid farmers in crop monitoring. [4] Senthilkumar and Jayanthy suggested a CNN-based model with novel activation and optimization, using K-means clustering to determine fertilizer consumption and increase crop output. [5] Nagaveni and Raghavendra compared machine learning approaches, with CNN outperforming other models in diagnosing plant diseases and determining fertilizer usage. [6] Ramesh and Hebbar employed Random Forest and HOG features for leaf classification, harnessing machine learning and publicly available datasets for scalable plant disease diagnosis. [7] Jayswal and Chaudhary compared traditional, machine learning, and deep learning methods, finding CNN to be the most effective in recognizing and categorizing plant diseases. [8] Li, Zhang, and Wang proposed using deep learning to treat plant diseases, addressing challenges in disease detection and early labeling of datasets. [9] Albattah and Nawaz developed a Custom CenterNet framework using DenseNet-77, achieving reliable plant disease classification and outperforming existing techniques. [10] Arsenovic and Karanovic focused on overcoming limitations in plant detection, proposing PlantDiseaseNet with 93.67% accuracy for disease diagnosis in various environments. [11] Baol and Zeng developed C3D-U-Net, a 3D COVID-19 segmentation model with intact encoding and local attention mechanisms, outperforming current techniques. [12] Kang and Wu utilized CNNs, including MSMA-U-Net, for efficient hippocampus segmentation in brain MRI, demonstrating improved feature extraction and faster segmentation. [13] Murmu and Kumar proposed a modified 3D U-Net CNN for medical image classification and segmentation, achieving high IOU and F1 scores on CT and MRI images. [14] Niu and Lin designed a Nested U-Net model with EfficientNet as the backbone for pneumothorax identification, showcasing superior segmentation performance. [15] Jia and Shi proposed a novel U-Net model with self-attention and deformable convolution for persimmon leaf disease segmentation, achieving improved accuracy in lesion detection.

# Problem Statement

The biggest threats that food security currently faces are pests along with crop diseases and the lack of necessary infrastructure to identify them. Oftentimes, the diseased crops are either misdiagnosed or go unnoticed completely until it’s too late. The brunt of this issue is mostly faced by the farmers whose entire livelihoods depend on the sale of crops. Although various efforts have been enforced to battle crop loss due to diseases such as the development of local plant clinics, the creation of new and improved pesticides, and so on, the knowledge and accessibility becomes reason for delay in cure. Nowadays smartphones and the internet on the other hand have become more and more accessible to remote areas. Most of them are equipped with quality cameras and other helpful features. Hence, we aim to create an application that will help farmers identify whether their tomato crop is diseased or not. We believe automation of the disease detection process would remove the scope of mistakes made by the farmers in detecting the disease and would lead to better crop yield overall.

# Background Study

**U-Net**

The U-Net architecture is also referred to as "U-Net" due to its U-shaped encoder-decoder architecture. Encoders capture incoming image data, while decoders reconstruct segmented output. U-Net encoder convolutional layers reduce the spatial dimensions of the input image while extracting usable features. It is similar to zooming out in order to observe the leaf's characteristics. U-Net decoders are opposite to encoders that reconstruct segmented output using the encoder's characteristics and increases spatial dimensions progressively. Skip connections in U-Net permit information transmission from encoder to decoder at the same spatial resolution that preserves vital data and enables the network to combine low-level and high-level characteristics. U-Net recalls the input characteristics while assessing the image's context. In order to identify unique features, the U-Net requires a compilation of input images with masks. The foliage masque displays both diseased and healthy portions by comparing its predicted output to the ground truth mask and adjusting its internal parameters. U-Net segments the leaf that reduces the gap between its predictions and the segmentation. After training, U-Net can identify unique features in images by segmentation and mask techniques. A U-Net trained using leaf images and masks can identify and segment problematic regions in preprocessed images, aiding in the diagnosis and management of unique features.

Contracting Path:

• Output feature map size: O = (W - F + 2P) / S + 1

(O: output size, W: input size, F: filter/kernel size, P: padding, S: stride)

Expanding Path:

• Output feature map size: O = (W - F) x S + F

(O: output size, W: input size, F: filter/kernel size, S: stride)

**Convolutional Neural Network**

CNNs are AI algorithms used for image categorization, object recognition, and pattern recognition that are based on the human visual system. Hierarchically analyzing the images, CNN layers extract pertinent characteristics whose first stratum is dominated by edges and lines, Further into the network the layers combine these fundamental qualities to detect shapes and textures. The network makes a prediction about the current image label based on these learned attributes. CNN gains the ability to spot recurring patterns in images during training. After that, a filter or kernel is moved across the input image to create a convolution. The filter multiplies its values by the image's pixels to create a local pattern feature map. At each convolutional layer, CNN applies many filters to extract different features. Pooling layers come after convolutional layers. These layers minimize the spatial dimensions of the features, enhancing the network's efficiency and allowing it to concentrate on the most important information. Following convolutional and pooling layers, CNN feeds data through fully connected layers that establish the feature classes. To reduce the difference between its anticipated outputs and the actual labels, the CNN adjusts the layer weights and biases throughout training. CNN gains the ability to distinguish between patterns in similar images with a variety of training examples.

Convolutional Layer:

• Output feature map size: O = (W - F + 2P) / S + 1

(O: output size, W: input size, F: filter/kernel size, P: padding, S: stride)

• Number of parameters: P = (F \* F \* D\_in \* D\_out) + D\_out

(D\_in: number of input channels, D\_out: number of output channels)

Pooling Layer (Max/Average pooling):

• Output feature map size: O = (W - F) / S + 1

(O: output size, W: input size, F: pooling size, S: stride)

Fully Connected Layer:

• Number of parameters: P = (D\_in \* D\_out) + D\_out

(D\_in: number of input neurons, D\_out: number of output neurons)

# Methodology

By leveraging the strengths of U-Net for image feature extraction, our goal is to create an advanced and artificially intelligent model capable of accurately identifying and classifying tomato leaf diseases. This model will empower farmers by providing them with a fast and reliable tool to determine whether their crops are infected, enabling them to take appropriate actions in a timely manner. The key advantage of our approach lies in the integration of deep learning techniques, which will revolutionize crop disease detection. By simply uploading an image of a tomato leaf, our model will not only detect the presence of disease but also classify the specific type of disease. This eliminates the subjective and inefficient nature of decision-making based solely on experience. The utilization of advanced deep learning algorithms will overcome the limitations of existing methods, such as the challenges posed by uneven backgrounds, the difficulty in accurately identifying and classifying diseases, and the lack of comprehensive datasets that reflect real-world scenarios. Our work includes a comprehensive discussion on image processing techniques, the utilization of CNN and U-Net architectures, dataset preparation methodologies, model training strategies, and rigorous evaluation procedures.

|  |
| --- |
|  |

Figure 1: Framework Diagram

**Dataset discussion**

The "Tomato Leaf Detection" dataset from Kaggle is the one used for the classification of tomato diseases. With respect to tomato plant age, we have selected plants that are in the middle to late phases of development. As a plant matures, leaf diseases frequently become more visible and distinct. Younger plants may not display distinct symptoms or have fully matured leaves, making disease detection more difficult. However, we have also ensured that the plants are not too old, as they may show evidence of multiple diseases or have leaves that are too damaged, which could impact the accuracy of our model. We must capture images of tomato leaves in a consistent and standardized orientation. For instance, capturing images with the leaf oriented in a consistent direction (e.g., with the top facing upwards) can aid in maintaining data consistency. Position the leaf to maximize the visibility of disease symptoms, without excessive overlap or obscuration. This helped to provide a more robust data set for training CNN models to detect diseases from multiple perspectives. Utilizing a high-resolution camera or a smartphone with a decent camera to capture images of leaves that are crisp and detailed. To enhance the visibility of disease symptoms, ensure that the leaf is well-focused and adequately illuminated. The data has different types of diseases for tomato leaves. The different types of diseases in the dataset are listed below:

* Tomato Mosaic Virus
* Target\_Spot
* Bacterial\_spot
* TomatoYellowLeafCurlVirus
* Late\_blight
* Leaf\_Mold
* Early\_blight
* Spidermites Two-spottedspider\_mite
* Tomato healthy
* Septoria Leaf Spot

Figure 2: Tomato Leaf Diseases

## Data Preprocessing

The work focuses on developing a robust data preparation pipeline for a computer vision-based system designed to analyze leaf images. The aim is to preprocess RGB leaf images and extract pertinent information to facilitate subsequent analysis and classification tasks. To ensure accurate and reliable results, a multi-step data preprocessing approach is employed. Initially, the RGB images are transformed into grayscale representations, reducing them to a single-channel format. This simplification step prepares the data for further processing stages. To enhance the image quality and remove unwanted noise, a Gaussian filter is applied. This effectively smoothes the grayscale images, resulting in cleaner and more refined representations. The Otsu thresholding technique is subsequently employed to convert the grayscale images into binary representations. By automatically determining an optimal threshold value, this technique effectively separates the foreground (leaf) from the background, facilitating subsequent analysis. To address any small holes or gaps in the foreground, morphological transformations are applied. These transformations, such as dilation and erosion, help to fill in gaps and create a more connected and complete representation of the leaf. By performing a bitwise AND operation between the binary images and the original color images, RGB images specifically segmented to highlight the leaf regions are obtained. This segmentation process successfully isolates the leaf from the background, enabling focused analysis.

Following the image segmentation stage, a comprehensive set of leaf characteristics is extracted, encompassing shape, texture, and color properties. Contours are employed to extract shape-related features, such as the leaf's area and perimeter, providing quantitative measurements for subsequent analysis. Color characteristics are evaluated by estimating the mean and standard deviation of each channel (Red, Green, Blue) in the RGB images. This analysis provides insights into the distribution and variability of color values within the leaf, enabling detailed color-based analysis. To further investigate the greenness of the leaf, the images are converted to the HSV color space. By computing the ratio of pixels with specific hue (H) channel intensity falling within a predefined range, an objective measure of the leaf's greenness is obtained. Lastly, to focus specifically on non-green regions of interest, the green color component is subtracted from the RGB images. This process reveals other color components and potential abnormalities or features of interest, allowing for more comprehensive analysis. Overall, the proposed data preparation pipeline encompasses various pre-processing steps to ensure accurate and informative representations of leaf images. These pre-processed images serve as a solid foundation for subsequent analysis and classification tasks in the context of plant pathology research.

# *Import libraries*

# *Set train\_dir and test\_dir for the directory paths of training and testing data*

# *Define a function get\_files(directory) to count the number of images in a directory*

# *If the directory does not exist, return 0*

# *Initialize count to 0*

# *Iterate over each subdirectory and file in the directory*

# *Increment count for each file found*

# *Return count*

# *Get the number of classes by counting the subdirectories in train\_dir*

# *Get the number of train images by calling get\_files(train\_dir)*

# *Get the number of test images by calling get\_files(test\_dir)*

# *Print the number of classes, train images, and test images*

***Use ImageDataGenerator to rescale the pixel values of images in the training set (1/255)***

***Augment the training data with transformations like shear, zoom, and horizontal flip***

***Split the training data into a validation set (20% of the data)***

***Use ImageDataGenerator to rescale the pixel values of images in the testing set (1/255)***

***Set the image width and height to 256***

***Set the input shape to (img\_width, img\_height, 1)***

***Set the batch size to 32***

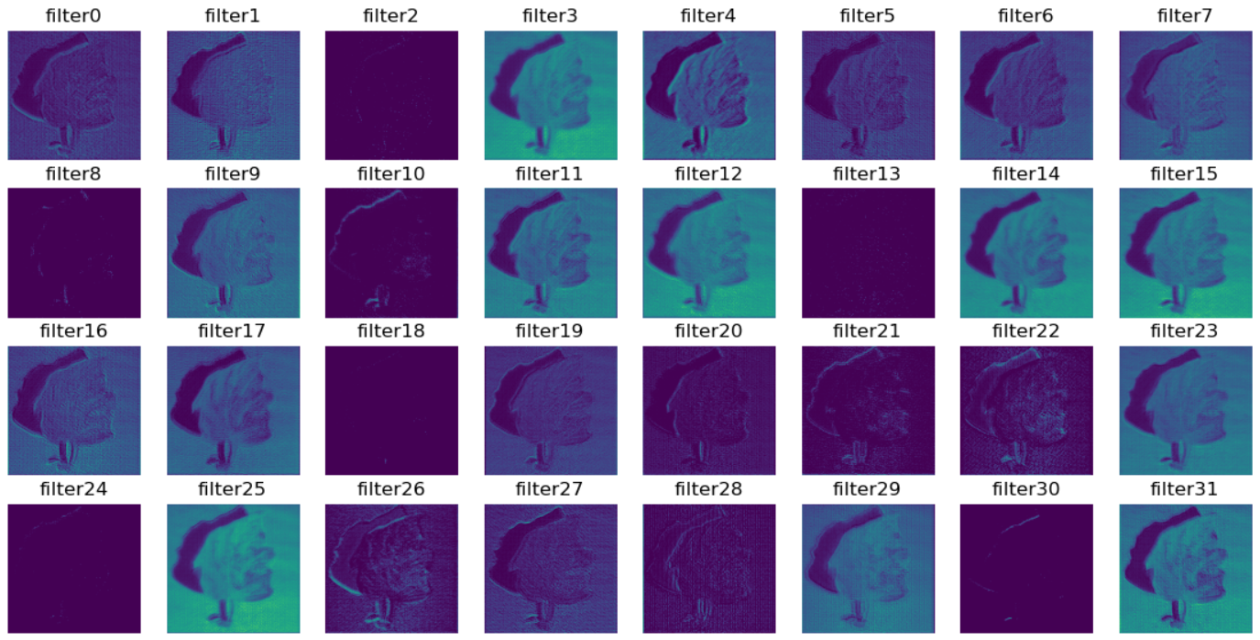
## In our research paper, we employed Convolutional Neural Network (CNN) architecture with 32 filters in the convolutional layer (Figure 3). These filters, also known as kernels, are small matrices used to derive distinctive features from input images. By executing convolutions on the input data, they play a crucial role in feature extraction. During the convolutional operation, each kernel traverses the input image and computes dot products between its weights and the corresponding pixel values in the receptive field. This procedure generates a feature map that highlights particular patterns or features in the input image.

## These kernels served as feature extractors for tomato leaf disease detection, enabling the network to learn and detect pertinent patterns, textures, and shapes associated with various types of diseases. By utilizing multiple kernels in each convolutional layer, the network was able to capture a vast array of features at various spatial scales. The CNN was taught to optimize the kernel weights in order to maximize the discrimination between healthy and diseased leaf samples. This adaptive learning enabled the network to become increasingly adept at recognizing disease-related features, thereby enhancing the classification accuracy of diseases.

## Using 32 filters in the convolutional layer was a strategic decision made to guarantee the network's ability to capture a wide variety of pertinent input data features. Each filter was designed to identify particular visual patterns or characteristics, such as edges, corners, textures, or higher-level characteristics such as object elements and shapes. By having multiple filters, the network could simultaneously learn and extract various types of features, thereby enhancing its ability to comprehend intricate data patterns.

## Using backpropagation and gradient descent algorithms, the weights of these filters were learnt and adjusted during the training phase. This learning process enabled the network to autonomously react to the data and extract its most informative characteristics. Visualizing the outputs of these 32 filters revealed valuable information about the network's learned representations. By analyzing the feature maps generated by each filter, we acquired a greater comprehension of the specialized categories of features that each filter was able to detect.

## Utilizing these 32 filters in the convolutional layer improved the network's ability to perform tasks including disease classification, object detection, and segmentation. Effectively identifying disease-specific patterns, the learned kernels aided in the early diagnosis and management of plant diseases. The contribution of CNN kernels to the development of an efficient and automated system for tomato leaf disease detection demonstrates the significance of feature extraction to the success of our research.



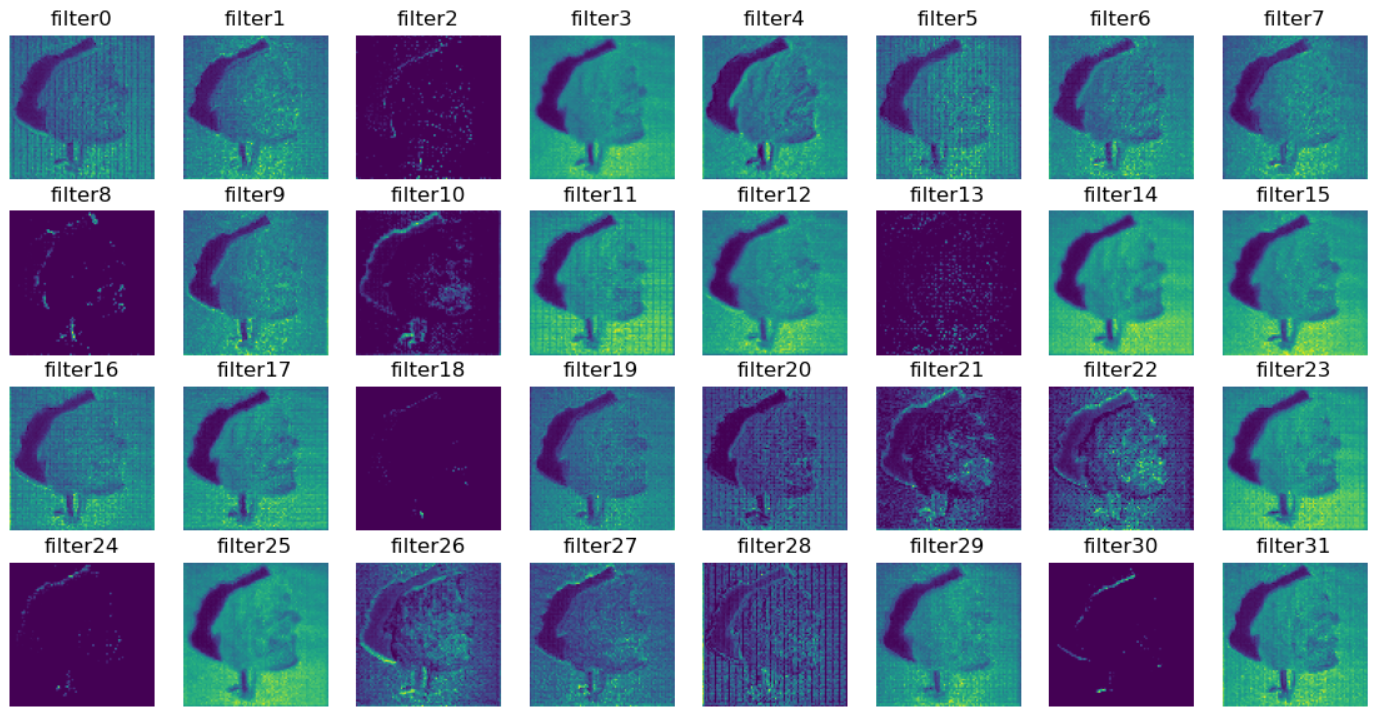


Figure 3: Visualization of CNN Filters and their output

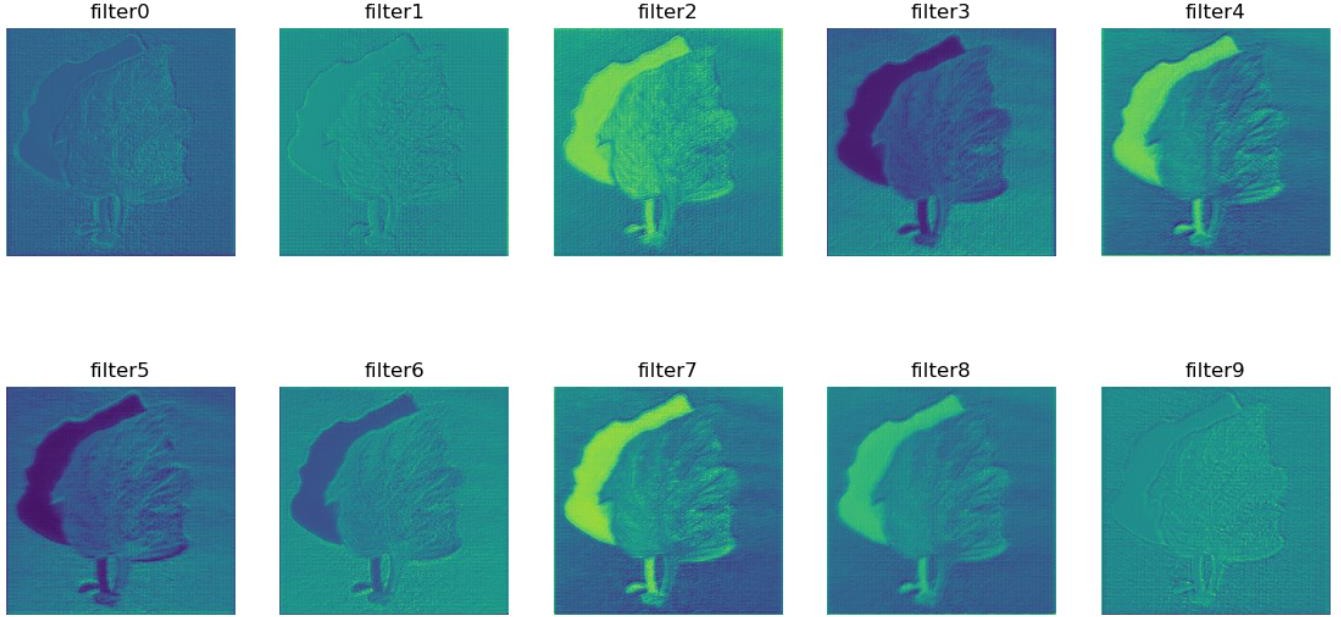
## Feature Extraction

This research applies feature extraction techniques to segmented input images in order to improve the accuracy and efficacy of the learned models. The objective is to extract informative and discriminative features that effectively convey the distinctive characteristics of various tomato leaf diseases. Various feature extraction methods, such as GLCM (Gray-Level Co-occurrence Matrix) and Texton features, are employed to accomplish this. GLCM measures the statistical properties of pixel intensity relationships, providing insight into the foliage regions' texture and spatial patterns. Texton features, on the other hand, capture local texture information through the use of filter banks, facilitating the discrimination of various disease patterns based on their texture signatures. By incorporating these feature extraction techniques, the work seeks to enhance disease detection by providing more representative and pertinent data to the U-Net CNN models. These extracted characteristics serve as valuable inputs for later phases of disease classification and localization. The extracted features facilitate the identification and differentiation of various plant diseases, allowing for early diagnosis and timely management interventions.

**GLCM Feature**

By incorporating GLCM features into our methodology, we improve the CNN and U-Net model ability to learn and differentiate between distinct disease types based on their distinct texture patterns. These extracted features provide valuable inputs for precise disease detection and facilitate timely disease management interventions. To derive GLCM features, we build a symmetric matrix in which each element represents the probability that a specific pair of gray-level values will occur. From this matrix, statistical measures such as contrast, homogeneity, entropy, and correlation are derived. These measurements capture vital textural information and serve as distinguishing characteristics for disease classification.

|  |  |  |
| --- | --- | --- |
| Sr. No. | GLCM Feature | Formula |
| 1. | Contrast | ∑𝑁−1 𝑃*i,j(i-j)2*  𝑖,𝑗=0 |
| 2. | Correlation | 𝑁−1  (𝑖 − µ𝑖)(𝑗 − µ𝑗)  ∑ 𝑃𝑖,𝑗 [ ]  √(𝜎𝑖)2(𝜎𝑗)2  𝑖,𝑗=0 |
| 3. | Dissimilarity | ∑𝑁−1 Pi,j |i - j|  𝑖,𝑗=0 |
| 4. | Energy | ∑𝑁−1 P*2i,j*  𝑖,𝑗=0 |
| 5. | Entropy | ∑𝑁−1 Pi,j(-ln Pi,j)  𝑖,𝑗=0 |
| 6. | Homogeneity | 𝑁−1  𝑃𝑖,𝑗  ∑ [ ]  1 + (𝑖 + 𝑗)2  𝑖,𝑗=0 |
| 7. | Mean | µi = 𝑁−1 i,j j = 𝑁−1 j(Pi,j)  ∑ i(P ) , µ ∑  𝑖,𝑗=0 𝑖,𝑗=0 |
| 8. | Variance | σi2 = ∑𝑁−1 Pi,j(i-µi)2 , σj2 = ∑𝑁−1 Pi,j(j-µj)2  𝑖,𝑗=0 𝑖,𝑗=0 |
| 9. | Standard Deviation | 2 2  σi = √σi , σj = √σj |



***Figure 4: Filter Map and Feature Visualization***

The extracted image is then made to undergo through a number of different filters.Above image (Figure 4) showcases the application of 10 filters on an input image. These filters, also referred to as kernels, are crucial to feature extraction. Kernels are compact matrices that conduct convolutions on the input image to extract distinct features. Using a variety of filters, the network captures a vast array of features at various spatial scales, allowing it to detect patterns, textures, and shapes associated with various types of tomato leaf diseases. It is essential to note that in actuality, a substantially larger number of filters are applied than those shown in the image.

Nonetheless, this image functions as a representative illustration of the use of kernels in feature extraction. The visualisation of these filters offers valuable insights into the learned representations and facilitates comprehension of the intricate patterns and structures detected by the network during the feature extraction process.

**U-Net Architecture based Tomato Leaf Segmentation**

U-Net has been widely used for biological applications such as detecting cancer, renal diseases,and tracking cells, among others. U-net has shown to be a very effective segmentation method in limited data circumstances (less than 50 training samples in some cases). A further benefit of using a U-net is that it lacks completely linked layers and, hence, has no input picture size restrictions. This feature enables us to extract features from images of varying sizes, which is a desirable characteristic for applying deep learning to high-fidelity biomedical imaging data. U- net is a great contender for image segmentation jobs due to its capacity to operate with a little amount of data and lack of input picture size requirements. The preprocessed tomato leaf image is passed through kernel/filters of 3X3 size and the number of nodes in each layer is increased exponentially starting with 16 nodes in the first hidden layer and subsequent layers contain 32, 64 and 128 and after this layer the number of nodes are decreased exponentially and the last layer gives the same number of nodes as in the beginning as the further CNN model has to be fed with the appropriate dimension of data. In each layer ReLU activation function has been used to avoid overfitting. On some dataset, if overfitting of data is encountered, the complexity of the model is to be reduced either by reducing the number of hidden layers or by reducing the number of nodes in each layer. During the development of this work, dropout function is also used to whether check if reducing the complexity of the current model can lend us some more better results but it came out to be of no use. Dropout function randomly select units and drop them from the model so that the complexity of the model is reduced to some extent.

**Tomato Disease Classification using Convolutional Neural Network**

In the input layer, distinct tomato leaf kinds images are inputted. In this case, there is no padding around the tomato leaf pictures. This allowed us to acquire a same input size on the feature map. Consequently, the characteristics of the complete image input may be handled without significant loss of information. The padded input image is then applied convolution filters, also known as kernels, to accomplish the convolution process. The convolution filter size is 3 x 3 with 1 pixel steps, and 32 distinct convolution filters are used to both convolution layers. The result of the convolution is then input into the activation function. In order to incorporate non-linearity into the network, an activation function is utilized since it can identify very complicated patterns. The two-layer curve procedure employs a wastrel activation function (Rectified Linear Unit). The feature maps have been filtered using maximum pooling. In the flattening layer, the retrieved tomato leaf image characteristics are flattened. They are then applied to the dense layer, also referred to as the completely linked layer.

***Add a multi\_U-Net\_model() layer***

***Add a Conv2D layer with 32 filters, a kernel size of (5, 5), and relu activation***

***Add a MaxPooling2D layer with pool size (3, 3)***

***Add a Conv2D layer with 32 filters, a kernel size of (3, 3), and relu activation***

***Add a MaxPooling2D layer with pool size (2, 2)***

***Add a Conv2D layer with 64 filters, a kernel size of (3, 3), and relu activation***

***Add a MaxPooling2D layer with pool size (2, 2)***

***Add a Flatten layer***

***Add a Dense layer with 512 units and relu activation***

***Add a Dropout layer with a rate of 0.25***

***Add a Dense layer with 128 units and relu activation***

***Add a Dense layer with num\_classes units and softmax activation***

***Print the model summary***

***Compile the model using Adam optimizer with learning rate 0.001, categorical cross entropy loss, and accuracy metric***

**Results and Discussion**

The data set used has 10,000 images and is further split into two sets consisting of 9,000 images as training set and 1,000 as testing set. we have successfully build and tested a machine learning model using U-Net Convolutional to classify different tomato plant leaf disease among the set of 10 diseases and we achieved an accuracy of 94.85% on validation dataset and 83.00% on randomly selected set of testing dataset.

To evaluate the performance of our proposed model, we conducted a series of experiments using carefully prepared datasets. These datasets include a training dataset, a testing dataset, and a randomly generated dataset. The results proposed validate the potential of our model to revolutionize the field of crop disease detection and provide valuable insights to farmers, ultimately benefiting the agricultural sector and ensuring food security for the population.

U-Net model showed more CPU utilization and took way more time to get trained than the normal model. The normal model took 6 hours to get trained over the 9000-image dataset but the u-Net model took approximately 24 hours (a day) to complete its training.

## Table 1: Time comparison between SVM, KNN,CNN and U-Net CNN models

|  |  |  |
| --- | --- | --- |
| Algorithm | Average CPU Utilization | Average GPU Utilization |
| SVM | 82% | 25% |
| K-NN | 70.3% | 20.1 % |
| CNN | 20% | 70% |
| CNN & U-NET | 65% | 30% |

Further we examined the performances of the two algorithms on some parameters like precision, recall, f1-score, support, latency, Cumulative Gain, and accuracy and we found the performance of U-Net model was better than the normal image processing.

Table 2: Performance metrics comparison chart

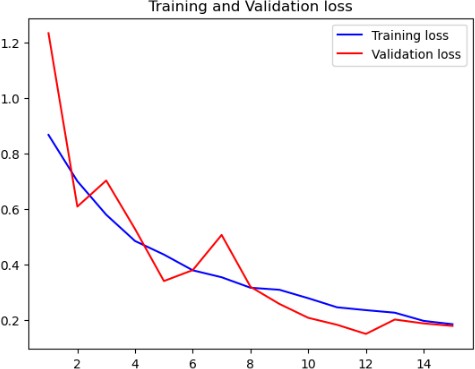
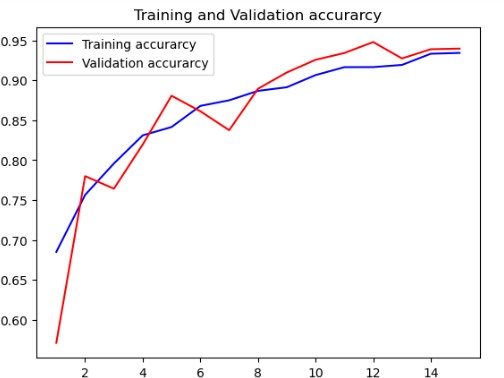
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F1-  Score | Support | Latency (s) | Loss | Frames per second | Accuracy |
| CNN | 0.10 | 0.10 | 0.10 | 100 | 21.294 | 43.82% | 6 | 87.19% |
| CNN & U-NET | 0.10 | 0.09 | 0.09 | 100 | 86.449 | 30.83% | 11 | 90.20% |

The U-Net model gave better results with the same number of epochs and steps and gave more validation and training accuracy of 94.85% and the loss is less in U-Net model as compared to the model implemented without U-Net.

## Table 3: Training and Validation Process Result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Epoch | Steps per epoch | Training Accuracy(%) | Training Loss (%) | Validation Accuracy(%) | Validation Loss(%) |
| SVM | 5 | 312 | 80.96 | 35.36 | 84.87 | 30.50 |
| 10 | 85.86 | 20.16 | 90.90 | 18.52 |
| 15 | 90.61 | 10.89 | **90.37** | 18.08 |
| CNN | 5 | 312 | 85.96 | 40.36 | 89.87 | 28.50 |
| 10 | 91.86 | 24.16 | 95.90 | 12.52 |
| 15 | 94.61 | 15.89 | **95.37** | 13.08 |
| U-Net CNN | 5 | 312 | 81.05 | 53.85 | 87.13 | 37.20 |
|  | 10 |  | 89.22 | 31.65 | 93.72 | 17.41 |
|  | 15 |  | 92.77 | 20.47 | **94.85** | 14.38 |

Furthermore, we have plotted accuracy on training and validation datasets as well as training and validation loss during the training process. We randomly selected a set of images from validation dataset and tested our model on that small dataset and we got an accuracy of 83.00%.



*Figure 2: Validation accuracy and loss graphs*

**Conclusion:**

This study has revealed the significant impact of plant diseases and pests on the agricultural sector in India, leading to substantial crop losses and adversely affecting the livelihoods of numerous farmers. To address this issue, various methods for estimating the prevalence of plant diseases were explored, but the U-Net Convolutional Neural Network (CNN) proved to be the most promising techniques. Traditional approaches to disease detection and classification suffer from subjectivity, inefficiency, and various limitations associated with current machine learning models. However, this work has developed a deep learning model employing U-Net and CNN that can accurately and efficiently identify and classify crop diseases. This AI model has the potential to significantly enhance the agricultural sector in India by increasing crop yields and supporting the livelihoods of farmers while also contributing to the country's economic growth.In future work we tend to make web page or app on which farmer can upload the image and further we can also be use it to set reminders to water any plant or fertilize it. The app can take in the live location and will analyze the weather and soil data for the location to best understand and prescribe the type of cultivation to the farmers. This will hugely benefit the farmers and ease their daily lives. Further, a community of farmers can be built to connect with each other better.

**References**

1.Khirade, S. D., & Patil, A. B. (2015, February). Plant disease detection using image processing. In 2015 International conference on computing communication control and automation (pp. 768-771). IEEE.

2.Sardogan, M., Tuncer, A., & Ozen, Y. (2018, September). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In 2018 3rd international conference on computer science and engineering (UBMK) (pp. 382-385). IEEE.

3.Shrestha, G., Das, M., & Dey, N. (2020, October). Plant disease detection using CNN. In 2020 IEEE Applied Signal Processing Conference (ASPCON) (pp. 109-113). IEEE

4.Yadhav, S. Y., Senthilkumar, T., Jayanthy, S., & Kovilpillai, J. J. A. (2020, July). Plant disease detection and classification using cnn model with optimized activation function. In 2020 international conference on electronics and sustainable communication systems (ICESC) (pp. 564-569). IEEE.

5.Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In 2019 5th International conference on advanced computing & communication systems (ICACCS) (pp. 281-284). IEEE.

6.Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., & Vinod, P. V. (2018, April). Plant disease detection using machine learning. In 2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C) (pp. 41-45). IEEE.

7.Jayswal, H. S., & Chaudhari, J. P. (2020). Plant leaf disease detection and classification using conventional machine learning and deep learning. International Journal on Emerging Technologies, 11(3), 1094-1102.

8.Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683-56698.

9.Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2022). A novel deep learning method for detection and classification of plant diseases. Complex & Intelligent Systems, 1-18.

10.Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. Symmetry, 11(7), 939.

11.Bao, Y., Zeng, H., Zhou, C., Liu, C., Zhang, L., Qian, D., ... & Lu, H. (2021, November). C3D-UNET: a comprehensive 3D Unet for Covid-19 segmentation with intact encoding and local attention. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 2592-2596). IEEE.

12.Lin, L., Kang, W., Wu, Y., Zhao, Y., Wang, S., Lin, D., & Gao, J. (2021, July). A 3D Multi-Scale Multi-Attention UNet for Automatic Hippocampal Segmentation. In 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC) (pp. 89-93). IEEE.

13.Murmu, A., & Kumar, P. (2021, December). Deep learning model-based segmentation of medical diseases from MRI and CT images. In TENCON 2021-2021 IEEE Region 10 Conference (TENCON) (pp. 608-613). IEEE.

14.Niu, H., Lin, Z., Zhang, X., & Jia, T. (2022, May). Image Segmentation For pneumothorax disease Based On based on Nested Unet Model. In 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA) (pp. 756-759). IEEE.

15.Jia, Z., Shi, A., Xie, G., & Mu, S. (2022, April). Image segmentation of persimmon leaf diseases based on UNet. In 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP) (pp. 2036-2039). IEEE.