International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Multi-Class Deep Learning Approach for Tomato Leaf Disease Detection System

Anu*[1], Kamna Solanki[2], Amita Dhankar[3]

Submitted: 28/09/2023 **Revised**: 14/11/2023 **Accepted**: 26/11/2023

Abstract: Tomato leaf infections pose a prevalent risk to sustained tomato cultivation, impacting numerous producers on a global scale. The timely identification, management, and resolution of tomato leaf specificity are of utmost importance in fostering optimal growth of tomato plants and guaranteeing an abundant supply of tomatoes to meet the increasing global demand and provide food security. The utilisation of computer-assisted technology for the identification and diagnosis of diseases presents in plant leaves is currently widespread. This study utilises about 10,000 tomato leaf photos sourced from the PlantVillage standard library to perform object localization. This paper presents a proposed Deep Learning approach that demonstrates effectiveness in autonomously segmenting and detecting diseases in tomato plant leaves. This work employed an image processing approches for the purpose of pre-processing and segmentation, alongside a multi-class convolutional neural network, in order to categorise ten distinct categories of diseases affecting tomato plant leaves. The process of segregating the contaminated area from the unaffected regions of the image was accomplished by the utilisation of thresholding, Gaussian blur, and canny edge detection techniques. Subsequently, the features were extracted by employing a Convolutional Neural Network model. The underlying model is evaluated by using the activation functions Relu and Leaky_Relu. It has been observed that the performance of presented architectures is superior when utilising the leaky_relu activation function contrasted to the relu activation function. Specifically, the training accuracy achieved with relu is 92.76%, whereas with leaky_relu it is 95.08%.

Keywords: *Deep learning, Thresholding, Normalization, Edge Detection, Gaussian Blur*

1. Introduction

Agriculture has historically served as the predominant economic sector for the majority of individuals in India. The profound commodification of agriculture has had a substantial impression on the natural environs. The detection of plant diseases is a substantial challenge within the agricultural domain. The timely identification of diseases shows a vital importance in mitigating the dissemination of diseases across plant populations, hence averting significant economic ramifications. The significances of disease in plants can conform from slight symptoms to the complete devastation of entire plantations, resulting in significant repercussions for the agronomic sector [1]. Deep learning is a computational paradigm that integrates advanced data analysis techniques with image processing methodologies, yielding very precise outcomes. Currently, deep learning is being widely employed across many domains, including but not limited to object detection, signal and voice recognition, as well as biomedical picture classification and segmentation. The utilisation of deep learning techniques in the agricultural domain has become prevalent, particularly in the context of detecting and classifying plant diseases. The Convolutional Neural Network is widely regarded as the most effective technique among many deep learning approaches. Different convolutional neural network designs, including as AlexNet and GoogLeNet, are now being employed for the purpose of detecting and classifying plant illnesses [2]. This study presents a novel framework utilising deep models for the identification of tomato plant diseases through the analysis of tomato leaf photos. This methodology has the potential to assist farmers in the categorization of illnesses that impact tomato growing through the utilisation of leaf images, hence circumventing the need for expensive expert examination. The proposed methodology involved the preprocessing of images and the utilisation of segmentation techniques to achieve high levels of accuracy. Additionally, the gathered photos of the dataset were enhanced for the purpose of training. Subsequently, a DenseNet architecture was employed to train a model on a dataset consisting of tomato plant photos, with the objective of detecting instances of tomato disease based on leaf images.

This article is organized in the subsequent manner: Section 2 encompasses a comprehensive review of the existing methodologies pertaining to the subject matter. In Section 3, the methodology is expounded upon, encompassing the description of the dataset employed, the pre-processing technique utilised, and the segmentation techniques employed. Section 4 provides an overview of the feature extraction process and the experimental setup. Section 5 provides a comprehensive analysis of the data obtained from the proposed Convolutional Neural Network (CNN)

[1,2,3] *UIET,Maharishi Dayanand University,Rohtak,India*
* *Corresponding Author Email: Anukadian182315@gmail.com*

model, including a performance comparison between two different activation functions. The conclusions are provided in Section 6.

2. Literature Review

To mitigate the impact of intricate backgrounds on the identification of plant diseases and pests in images, a segmentation technique employing a complete convolutional network method is employed. This algorithm is based on the VGG-16 model. Subsequently, a refined Dual-Path Networks model is introduced with the aim of enhancing the efficacy of feature extraction process. The SNDPN model integrates the joining approach employed in Desnet as well as Resnet layers. It constructs a neural network by incorporating the SN layer and dynamically optimises the variable of the DPN through the activation of the normalised layer. This enhancement enhances the adaptability of the network to various disease and insect pest types, as well as improves the training efficiency of the network [3]. The Indian Agricultural Research Institute has gathered a dataset consisting of real-time images depicting the early blight disorders in tomatoes. The HDL models that were proposed demonstrated outstanding performance on the IARI TomEBD data, exhibiting a notable accuracy level. Moreover, the suggested approach has been validated using two publicly accessible datasets on plant diseases, specifically Plant Village-TomEBD [4].The application of deep learning techniques was employed to recognize and categorize different disorders affecting the foliage of tomato plants. The main aim of the presented work was to implement real-time execution of the deep learning algorithm using robotics. The robotics would possess the capability to identify plant disorders while navigating either manually or autonomously inside the field or greenhouse environment. Similarly, the identification of illnesses can be achieved by the analysis of high-resolution pictures captured by sensors integrated within artificial greenhouses. The selection of deep learning architecture had a pivotal role in the implementation process. Two distinctive Deep learning frameworks, namely AlexNet and SqueezeNet, were initially evaluated [5]. The implementation of a machine learning model that is most appropriate for detecting tomato crop illnesses in standard RGB photos has been accomplished. The study employed deep learning models, specifically DenseNet with layers count of 161 and 121, as well as VGG16 using transfer learning. This article utilises a collection of photos depicting various forms of infections caused by pest infestations and plant illnesses, specifically focusing on infected plant leaves that have been categorized into six distinct groups. The findings exhibited promising outcomes, indicating that DensNet161 achieved an accuracy rate of 95.65%, DensNet121 achieved an accuracy rate of 94.93%, and VGG16 achieved an accuracy rate of 90.58% [6]. Deep learning has played a crucial role in enhancing classification and detection tasks within the domain of precision agriculture. Nevertheless, it is imperative to shift towards the widespread implementation of these approaches using cost-effective and energy-efficient equipment that can be readily employed in everyday agricultural practices. This study presents the training assessment of four contemporary CNNs models for classifying disorders in tomato plants. The chosen models employ a depth-wise separable convolution architecture, which is specifically designed for utilisation in devices with low power consumption. Quantitative and qualitative evaluation and investigation are conducted using quality indicators and saliency maps. A graphical user interface (GUI) is designed for the enactment on the Raspberry Pi 4 micro-computer [7].The authors endeavoured to deploy the AlexNet modification architecture-based Convolutional Neural Network function on the Android frameworks aiming to predict tomato illnesses using leaf images. The prediction model was developed using a dataset consisting of 18,345 training data and 4,585 testing data. The optimal model, employing the Adam optimizer using learning rate of 0.0005, was trained for 75 epochs using a batch_size of 128. The model utilised a cross-entropy loss function without any compromises. It achieved a remarkable accuracy of 98%, demonstrating its effectiveness in accurately classifying the data. The model also exhibited a high level of precision, with the metrics include a rigidity rate of 0.98, a recall rate of 0.99, and a F1 score of 0.98. Furthermore, the model's loss was measured at 0.1331, indicating its ability to minimize errors during the training process. Overall, the classification results obtained from this model were highly satisfactory, displaying both good performance and precision [8].

3. Material and Method

This section offers a complete analysis of the approaches employed in the present analysis, encompassing both the structural aspects and the detection process. This study used the Convolutional Neural Network to quantify the condition. The utilisation of image processing techniques in the early detection of damaged leaves can aid farmers in effectively treating the affected foliage [9]. The process of detecting leaf disease is illustrated in Fig. 1. The entirety of the dataset falls under the scope of pre-processing and data augmentation. Several pre-processing practices, such as noise reduction and rescaling, are employed. Subsequently, data augmentation functions are applied to augment the number of images. The process of segmenting the input image involves the application of thresholding, Gaussian blur, and canny edge detection techniques. Subsequently, the entirety of the data is segregated into several sets for training and testing purposes. The suggested work utilises training samples, and the evaluation of its effectiveness is conducted following the experimentation with testing samples.

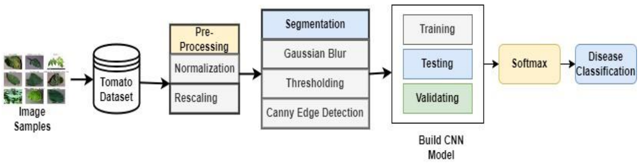

Fig. 1. Tomato Disease Detection Process

3.1. Dataset and Pre-processing

The information utilised in this study was obtained from the widely recognized Plant village dataset. This dataset comprises over 50,000 photos encompassing 14 distinct crop species and 26 different illnesses. We have opted to utilise a dataset consisting of 10,000 photos specifically focused on Tomato leaves. This dataset encompasses samples representing nine distinct types of tomato illnesses, in addition to healthy leaves. In total, there are ten classes within the dataset, namely bacterial spot, mosaic virus, late blight, leaf mold, target Spot, early blight, spider mites yellow leaf curl. The categorization technique involved the selection of two-spotted spider mite, Septoria leaf spot, and healthy samples. Fig. 2 depicts the image samples that have been gathered from the source located in the plant village. The image samples were downsized to dimensions of 150×150 in order to expedite computational processes, while ensuring that the integrity of the data remained unaffected. The photos of citrus fruit that have been captured undergo an initial pre-processing stage. The process of picture rescaling has been executed to generate new iterations of the photos. However, it should be noted that this particular procedure does not guarantee preservation of all information. It is imperative to improve the image to subjectively improve its clarity and address issues such as undesirable flickering, contrast deficiencies, and the potential for extracting additional information. The images utilized for model training undergo preprocessing procedures such as normalization and rescaling. These techniques ensure that the model receives consistent and standardized data for the purpose of training and validating the image dataset. Furthermore, the addition of data augmentation features serves to boost the capabilities of Convolutional Neural Network (CNN) models. Consequently, eight additional standards have been incorporated to supplement both our training and testing datasets. Researchers commonly employ a conventional augmentation system to expand the size of their input data samples that involves rotating the photos by 30 degrees horizontally shifting them by 50 pixels, and vertically shifting them by 30 pixels. Following the implementation of augmentation techniques, the initial collection of 10,000 tomato specimens, categorized into 10 distinct classes, experienced an expansion to over 30,000 specimens. The tomato leaf samples depicted in Fig. 3 illustrate the results obtained following the implementation of data augmentation techniques.

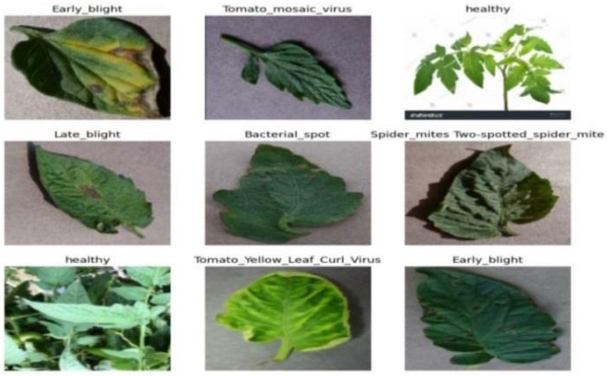

Fig.2 . Collected Image Samples of Tomtao Leaves

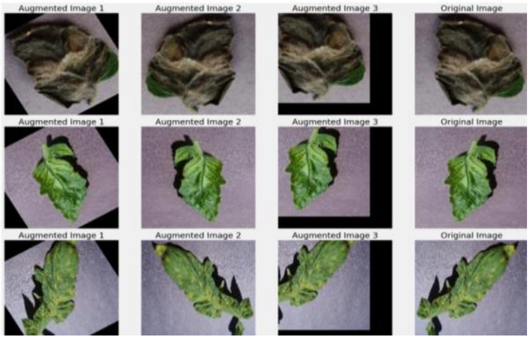

Fig. 3. Tomato Disease Image Samples after Augmentation

3.2. Segmentation

The segmentation procedure is of utmost importance in the study of image quality. The concept of image segmentation pertains to the computational procedure of dividing a digital image into numerous discrete sections, wherein each region has pixels that exhibit similar features. This study has employed three distinct segmentation algorithms, namely thresholding, Gaussian blur, and canny edge detection.

3.2.1. Thresholding

Thresholding is a segmentation technique utilised to produce a binary image, wherein pixels are assigned either a value of 0 or 1. This binary representation requires just one bit to retain pixel intensity. The process involves dividing a grayscale image into two distinct regions by employing a predetermined threshold value [10]. Therefore, pixels with intensity values exceeding the specified threshold will be classified as white or assigned a value of 1 in the resulting image, while all other pixels will be classified as black or assigned a value of 0. There exist two primary categories of pixels, with one group including pixels of a deeper hue and the other group consisting of pixels of a lighter hue. An object of interest may be positioned within the background. By employing a suitable threshold value of 128, the image may be effectively partitioned into two separate and distinguishable parts. Put simply, if a threshold T is given, the segmented picture ω(m,n) can be calculated using the following method as shown in equation 1:

$$\omega(m,n) = 1 \qquad if\ \varphi(m,n) > threshold(\delta) \qquad (1)$$

$$\omega(m,n) = 0 \qquad if\ \varphi(m,n) \leq threshold(\delta)$$

3.2.2. Gaussian Blur

The Gaussian blur aspect is achieved through the application of a Gaussian function to blur (or smooth) an image, resulting in a reduction of the noise level. The aforementioned filter can be regarded as a non-uniform low-pass filter that effectively preserves low spatial frequency components while simultaneously diminishing picture noise and insignificant details inside an image. The common approach involves the convolution of the images using a Gaussian kernel. The 2-dimensional usage of the Gaussian kernel is mathematically represented in equation (2)

$$\dot{G}_{2\ dim} mn\sigma = \frac{1}{2\pi\sigma^2} e^{\frac{-m^2+n^2}{2\sigma^2}} \qquad (2)$$

In the given context, σ represents the distribution of standard deviation, while m and n serve as the location indices. The parameter σ governs the dispersion of data points from the average value in the Gaussian distribution, hence influencing the degree of blurring observed around a given pixel. One potential method for reducing the noise present in the image is the application of Gaussian blurs in order to provide a smoothing effect [11]. In order to do this, the image convolution approach is employed using a Gaussian Kernel of various sizes, such as 3x3, 5x5, 7x7, and so on. The size of the kernel is contingent upon the anticipated blurring impact. In essence, a reduction in kernel size corresponds to a decrease in the perceptibility of blur. In this study, a Gaussian kernel size of 5 by 5 was utilized. The equation representing a Gaussian filter kernel of a specified size is as follows in equation (3).

$$\dot{G}_{2\ dim} = \frac{1}{2\pi\sigma^2} e^{\frac{-(m-(k+1))^2+(j-(k+1))^2}{2\sigma^2}} \qquad (3)$$

3.2.3. Canny Edge Detection

The utilisation of canny edge detection is a widely adopted technique for image segmentation, as it efficiently identifies the edges of an image while simultaneously reducing the presence of noise. The technique consists of five essential stages: noise_reduction, gradient_calculation, maximum_suppression, double_thresholding, and edge_tracking utilising hysteresis. The applications of canny edge detection techniques result in the efficient removal of edges present in the image, hence enhancing the quality of subsequent analysis. The process of gradient computation involves determining the intensity and direction of edges within an image through the utilisation of edge detection operators. Pixels with intensity values falling under the specified limits are designated as weak [12]. The Hysteresis process is employed to distinguish between strong pixels and those that are deemed irrelevant. According to the findings from the threshold analysis, the hysteresis phenomenon involves the conversion of low-intensity pixels to high-intensity pixels, but only if there is at least one neighbouring pixel that is already classified as high-intensity. Fig. 4 depicts the outcomes of three segmentation techniques employed on the image samples.

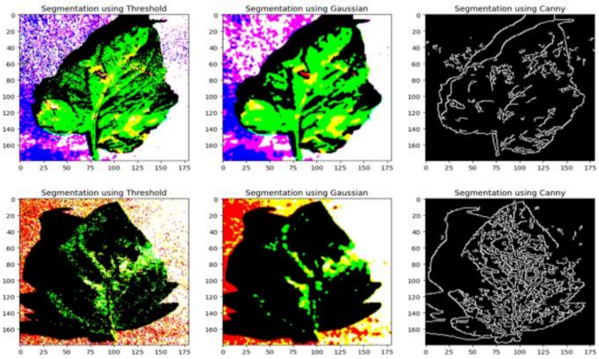
Segmentation using Threshold
Segmentation using Gaussian
Segmentation using Canny
Segmentation using Threshold
Segmentation using Gaussian
Segmentation using Canny
0
20
40
60
80
100
120
140
160
0
25
50
75
100
125
150
175

Fig. 4. Segmented Image Samples

Proposed Algorithm:

1. Read the RGB images (Img)
2. Implement Resizing (256 × 256)
3. Normalize pixel values to the range [0, 1]
4. Perform Rescaling (1.0 / 255)
5. Performed Data Augmentation using rotation (30◦), horizontal_shift (50 pixels), and vertical_shift (30 pixels)
6. Transform the images into grayscale
7. Implement a Gaussian blur to reduce noise
8. Perform edge detection using the Canny detector
9. Find contours in the edged image
10. Function to segment an image using different techniques threshold_value=128, kernel_size=5
11. Build a CNN model
12. Create an Adam optimizer with the specified learning rate
13. Evaluate model performance (confusion matrix, accuracy, precision, specificity, FScore, loss, recall)

4. Model Training

Convolutional networks draw inspiration from biological processes, as their neural connectivity pattern closely mirrors the organisational structure observed in the visual brain of animals. Cortical neurons exhibit selective responsiveness to stimuli within a limited area of the chromatic regions, commonly referred to as the amenable arena. The receptive areas of distinct neurons exhibit partial overlap, so collectively encompassing the entirety of the visual field. A Convolutional Neural Network is composed of an initial layer, a final output layer, and several intermediate hidden layers [13]. The concealed strata of a Convolutional Neural Network (CNN) often encompass convolutional strata, pooling strata, fully connected strata, and normalisation strata. The convention in neural networks is to describe the process as a convolution. From a mathematical perspective, it can be said that the operation in question is a cross-correlation rather than a convolution. The significance of this is limited to the indices within the matrix, determining the placement of weights at specific positions.

4.1. Experimental Setup & Feature Extraction

The experiment was implemented on a PC with windows 10 operating system, utilising the Google Colaboratory GPU runtime. The computer had access to 8 GB of storage space on Google Drive and operated on a 64-bit system. The training and validation methods were facilitated by employing the Keras 2.4.3 outline and utilising the backend as Tensorflow. The complete datasets were subsequently combined in order to train and evaluate the model. Each individual data point was carefully checked to ensure the elimination of misclassification. The training, validation, and test sets are composed of photos randomly chosen from the input dataset, with a ratio of 8:2 for each category. The training and validation sets are utilised for the training of network, while the test set is implemented to assess the outcome of the trained model. The neural networks are trained with Stochastic Gradient Descent optimizer with a momentum 0.9. The inclusion of the Rectified Linear Unit (ReLu) activation function and max pooling is seen. The suggested model incorporates an activation function that selectively includes positive gradients while excluding negative weights. Despite its notable achievements, particularly in the realm of deep networks, the rectified linear unit exhibits certain limitations. Firstly, it should be noted that the ReLU function is not continuously differentiable. The computation of the gradient is not possible at x=0. While the issue at hand may not be considered significant, it does have a minor influence on training performance. Additionally, the Rectified Linear Unit (ReLU) function assigns a value of zero to all input values that are less than zero. This phenomenon can be advantageous in cases of limited data availability [14]. However, it is imperative to remind that due to the fact that the gradient of 0 equals 0, neurons that reach significantly negative values are unable to recover from being trapped at 0. The neuron undergoes an effective cessation of function, leading to the phenomenon commonly referred to as the "dying ReLU" problem. This phenomenon can result in the network seeing a significant decline in its learning capabilities and hence performing below expectations. The Leaky Rectified Linear Unit (LReLU) aims to address these challenges by introducing a slight negative slope for negative input values within the ReLU utility. Equations (6) and (7) illustrate the Leaky Rectified Linear Unit (LReLU) and its corresponding derivative. A new activation function termed Leaky ReLu is applied in this work and compared with the model with basic Relu function [15]. The activation function employed in this study incorporates both negative and positive gradients, enabling the detection of regions of interest. The mathematical expression for the ReLu function is depicted from equations (4-5).

$$g(x) = \begin{cases} 0, \ \forall\, x < 0 \\ x, \ \forall x > 0 \end{cases} \tag{4}$$

$$\frac{d\,g(x)}{dx} = \begin{cases} 0, \ \forall x < 0 \\ \ \ 1, \ \forall x > 0 \end{cases} \tag{5}$$

$$g(x) = \begin{cases} \ x, \ \forall\, x > 0 \\ \alpha x, \ \forall x \leq 0 \end{cases} \tag{6}$$

Where α=0.01

$$\frac{d\,g(x)}{dx} = \begin{cases} 0.01, \ \forall x < 0 \\ \qquad 1, \ \forall x > 0 \end{cases} \tag{7}$$

$$\mathrm{g}\ (\ x\ ) = max(\ 0.01 \times y, y) \tag{8}$$

In this context, the vertical axis, denoted as the y-axis, indicates the relative importance or weightage assigned to input signals. On the other hand, the horizontal axis, referred to as the x-axis, shows the signals that have been activated or triggered. In the event that the input is positive, this method yields the value of y. Conversely, if the input is negative, it produces a far smaller value, specifically 0.01 times y. The output characteristics are transformed from a grid structure in the output layer to a linear format, allowing the layers to retain distinct outcomes from the prior layers. Subsequently, two densely connected layers are utilised, with the first layer accommodating an excessive count of hidden layers in correspondence to the output layer, and the second layer accommodating the prediction class's count. The initial dense layer, denoted as Dense (64), signifies the presence of 64 hidden layers or feature matrix. Subsequently, the subsequent Dense (10) layer indicates the existence of 10 distinct prediction classes pertaining to tomato leaves. The Adam optimisation methodology is a commonly employed method for iteratively improving network weights based on training data [16]. When confronted with intricate issues characterised by a substantial amount of data or variables, the approach exhibits remarkable efficacy and consumes minimal memory resources, as demonstrated in equation 9.

$$\begin{aligned} g_{t=}\beta_1 g_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right]\alpha_t \\ = \beta_2 \alpha_{t-1} \\ + (1-\beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2 \end{aligned} \tag{9}$$

Table 1: Configuration of proposed CNN model with Relu and Leaky Relu Activation Function

CNN with Relu Activation			**CNN with Leaky_Relu Activation**		
Layers	**Output Shape**	**Kernel Size**	**Layers**	**Output Shape**	**Kernel Size**
Conv2d	(178,178,32)	3×3	Conv2d	(178,178,32)	3×3
Relu	(178,178,32)	----	Leaky_relu	(178,178,32)	----
Max_pooling2d_2	(89,89,32)	2×2	Max_pooling2d	(89,89,32)	2×2
Conv2d	(87,87,32)	3×3	Conv2d	(87,87,32)	3×3
Relu	(87,87,64)	-----	Leaky_relu	(87,87,64)	-----
Max_pooling2d_3	(43,43,64)	2×2	Max_pooling2d_3	(43,43,64)	2×2
Flatten	(118336)	----	Flatten	(118336)	----
Dense_2	(64)	----	Dense	(64)	----
Relu	(64)	----	Leaky_relu	(64)	----
Dense_3(softmax)	(10)	----	Dense_3(softmax)	(10)	----

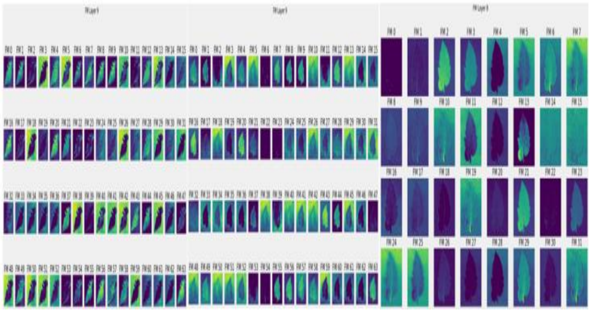

Fig. 5. Feature Map obtained using Relu Activation Function in Proposed Model

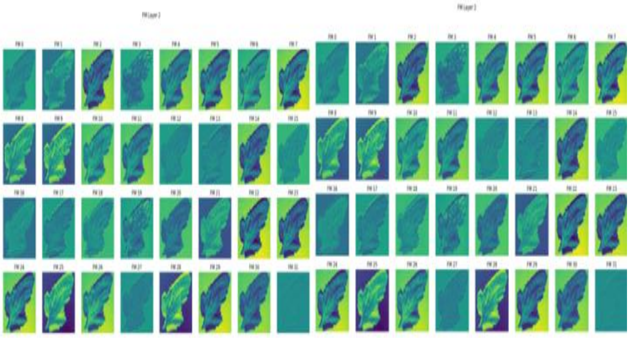

Fig. 6. Feature Map obtained using Leaky_Relu Activation Function in Proposed Model

5. Result Analysis

The efficiency of the proposed segmentation and classification model is evaluated using the following metrics:

5.1. Quantitative Analysis

Digital information is utilised to provide data and information in quantitative analysis, which involves the examination of significant data derived from digital images. Analysis of data is expressed in different terms, such as; True Positive, True Negative, False Positive, and False Negative. The term "Precision" denotes to the ratio of acceptably predicted positive values to the total number of values determined from the provided picture dataset, as denoted by Equation (9). The term "recall value" refers to the computation of the positive value in relation to the expected value of a whole class, as represented by Equation (10).

$$P\ recision = Tpositive\ Tpositive + Fpositive \quad (9)$$

$$Recall = Tpositive\ Tpositive + Fnegative \quad (10)$$

The term "Tpositive" denotes the accurate and genuine value, specifically referring to the identification of the infected region as the true infected area. The term "Tnegative" depicts the non-infected portion as accurately as the non-infected portion itself. The phenomenon of interest demonstrates the presence of an infected segmented portion within an otherwise healthy region. The phenomenon of false negatives is observed when an event in a picture, which is actually infectious, is incorrectly identified as being healthy. The F1-Score, as represented in Equation (11), is obtained by calculating the mean of recall and precision in order to assess the performance of infected regions within a given dataset.

$$F1 - Score = 2 \times P\ recision \times Recall\ P\ recision + Recall \quad (11)$$

5.2. Confusion Matrix

Classifiers may experience difficulty when confronted with a multitude of interconnected forms from several classes. Infected tomato leaf images may exhibit considerable intricacy across many levels or backgrounds, leading to diminished performance in discerning patterns within the same category. The utilisation of a confusion matrix allows for the visual confirmation of a model's accuracy in categorization [17]. The entire dataset utilised in our study is partitioned into two distinct sets, namely the training set and the testing set. To make the process of training and evaluating the model as efficient as possible, the division is done at random. The model's accuracy inside a specific class is closely linked to the level of colour saturation observed in the visualization results. In the context of predictive outcomes, it may be observed that successful forecasts tend to align in a diagonal pattern, while unsuccessful predictions exhibit a tendency to deviate from this diagonal alignment. The classification accuracy can be visually evaluated based on the provided data.

Table 2 displays the confusion matrix for the 10 courses in the final test. In this study, a total of 80 photos were utilised for developing the model, and the dataset was divided into 80:20 ratio. The model's quantitative analysis is characterised by four distinct terms: "True Positive (TP)", "False Positive (FP)", "True Negative (TN)", and "False Negative (FN)". The count of images is 562, depicting mosaic virus with corresponding Tpositive values. Additionally, there are 507 images illustrating target spot with Tpositive values. The number of images displaying Tpositive values for bacterial spot is 491, while the number of images for yellow leaf curl virus, leaf mould, early blight, spider mites, late blight, and healthy are 536, 457, 436, 572, 559, 534, and 443, respectively. Diseases such as Septoria leaf spot, leaf mould, early blight, target spot, bacterial spot, and tomato mosaic virus can be differentiated from other diseases based on their distinct symptoms and detection rates. Based on the analysis of the confusion matrix, it can be observed that the detection model exhibits a higher susceptibility to misinterpretation when distinguishing between the other classes. The accuracy rates of various diseases, such as Septoria leaf spot, target spot, leaf mould, early blight, bacterial spot, mosaic virus, Late Blight, Spider Mites, and Yellow Leaf Curl Virus, are reported to be 97.62%, 97.02%, 96.55%, 95.77%, 96.82%, 97.7%, 96.73%, 98.05%, and 97.75%, respectively, as depicted in Fig. 7. Additionally, the accuracy rate for healthy plants is recorded as 95.9%. Fig. 7 illustrates the accuracy, precision, recall, and F1-score attained by the proposed model for individual classes. Meanwhile, Fig. 8 & 9 showcases the accuracy of classification and loss patterns observed during the training and validation curves when utilizing the Relu activation function in the proposed model. Fig. 10 & 11 presents the accuracy of classification and loss patterns observed in the training and validation curves when employing the Leaky Relu activation function in the proposed model.

relation to the expected value of a whole class, as represented by Equation (10).

$$P\ recision = Tpositive\ Tpositive + Fpositive \quad (9)$$

$$Recall = Tpositive\ Tpositive + Fnegative \quad (10)$$

The term "Tpositive" denotes the accurate and genuine value, specifically referring to the identification of the infected region as the true infected area. The term "Tnegative" depicts the non-infected portion as accurately as the non-infected portion itself. The phenomenon of interest demonstrates the presence of an infected segmented portion within an otherwise healthy region. The phenomenon of false negatives is observed when an event in a picture, which is actually infectious, is incorrectly identified as being healthy. The F1-Score, as represented in Equation (11), is obtained by calculating the mean of recall and precision in order to assess the performance of infected regions within a given dataset.

$$F1 - Score = 2 \times P\ recision \times Recall\ P\ recision + Recall \quad (11)$$

Table 2: Confusion Matrix for Ten Tomato Leaves Classes

Class	**Mosaic viral nfIection**	**Target Spot Disease**	**Bacterial Spot**	**Yellow Leaf Curl Viral Infection**	**Late Blight Disease**	**Leaf Mould Fungal Infection**	**Early blight**	**Infestation of Spider Mites**	**Septoria Leaf Spot**	**Healthy**
Mosaic viral infection	562	6	5	7	9	23	17	2	2	29
Target Spot Disease	8	507	6	11	4	9	25	4	1	18
Bacterial spot	3	4	491	12	7	11	24	2	4	15
Yellow Leaf Curl Viral Infection	4	10	7	536	7	13	16	3	3	8
Late Blight Disease	2	12	21	9	534	28	21	2	10	25
Leaf Mould Fungal Infection	3	15	9	7	6	457	3	3	5	13

Early Blight	7	15	17	6	9	14	436	6	8	8
Infestation of Spider Mites	2	12	13	5	6	12	18	572	2	19
Septoria Leaf Spot	3	8	17	4	13	17	15	3	559	22
Healthy	6	11	14	3	5	16	25	3	6	443

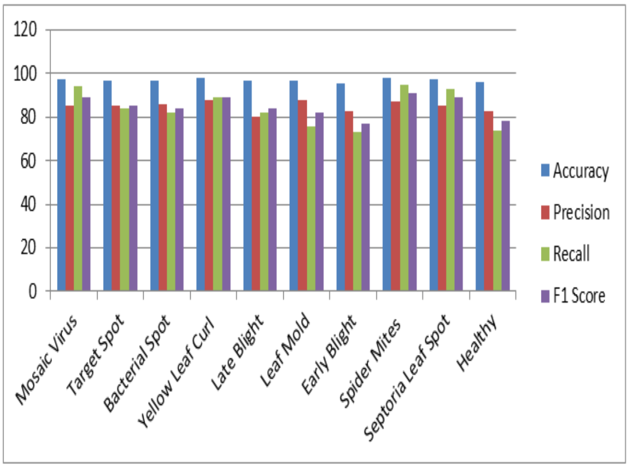

Fig. 7. Performance Results for Each Class of Tomato Disease

Table 3: Evaluation results of Multi-classification Model

Model	**Training Accuracy**	**Training Loss**	**Validation Accuracy**	**Validation Loss**
CNN with Relu	92.76	0.2415	99.45	0.0484
CNN with Leaky Relu	95.08	0.2007	99.75	0.0344

Table 4: Key Findings of Extant Literature

Ref.	**Defect Type**	**Classification**	**Technique**	**Accuracy**
[18]	External defects with stems/calyxes	Binary-Classification	ResNet50	94.6%
[19]	Mosaic virus, yellow leaf curl, and healthy images	Multi-Classification	Fused alex-SVM machine & convolution neural network	99.62%.
[20]	Immature Tomatoes	Binary-Classification	High-speed Region-Based Convolutional Neural Network Utilizing ResNet-101	87.83%
[21]	Early blight, healthy septoria_spot and late blight	Multi-Classification	Gray-Level Co-Occurrence Matrix and Support Vector Machine	100% for healthy, 95% for early blight, 90% for septoria and 85% for late blight
Proposed Work	10 Diseases Types	Multi-Classification	CNN with thresholding, Gaussian blur, and canny edge detection	99.75 Validation Accuracy

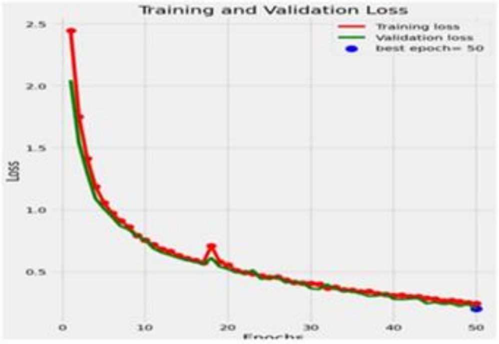

Fig. 8. Loss Curve using Relu Activation Function in Proposed Model

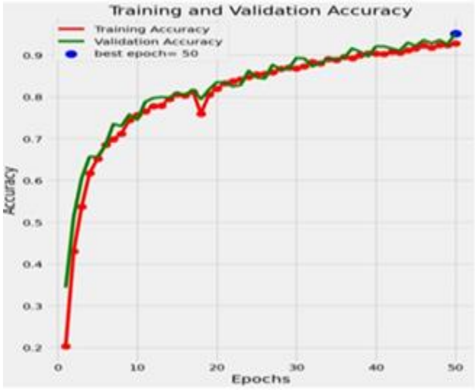

Fig. 9. Accuracy Curve using Relu Activation Function in Proposed Model

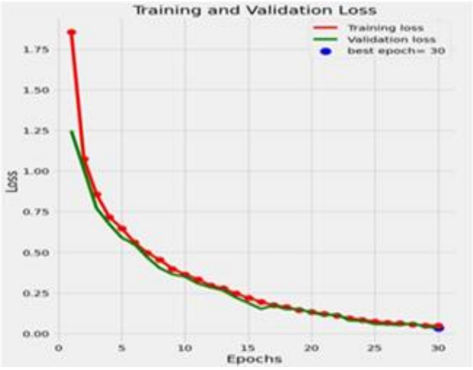

Fig. 10. Loss Curve using Leaky Relu Activation Function in Proposed Model

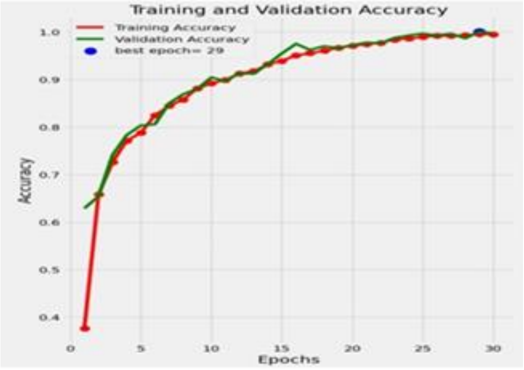

Fig. 11. Accuracy Curve using Leaky Relu Activation Function in Proposed Model

Table 3 presents the outcome results in relation to disease kind, classification type, technique employed, and attained accuracy. The suggested model demonstrates superior accuracy when compared to other current models.

6. Conclusion

Symptoms of infection can manifest in several anatomical regions of the plant, with leaves frequently serving as a primary indicator for identifying plant pathology. Scientists globally are utilizing advanced computer vision technology to carry out extensive experiments on the recognition of tomato plant disorders through the application of leaf image analysis techniques. In past decade, there has been extensive utilization of deep learning methodologies for the detection of plant diseases. The model efficiency was enhanced through the utilization of three segmentation techniques, including thresholding, Gaussian blur, and canny edge detection. This study proposes a diagnostic paradigm for tomato plant disorders based on the convolutional neural network (CNN) technique.

Data supplementation enhances the performance and generalization capabilities of the model. The findings of the proposed study demonstrate favorable outcomes in the identification of healthy and various types of affected leaves. The outcomes indicate that the proposed work has a notable level of precision. A comparative analysis was conducted between the relu and leaky_relu activation functions employed in the proposed deep learning model. The results indicate that the leaky Relu activation model exhibits superior accuracy in comparison to the ReLU activation function. In subsequent research, the suggested methodology can be extended to facilitate the early detection of various additional types of diseases prevalent in tomato plants.

References

[1] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In *2017 6th International conference on agro-geoinformatics* (pp. 1-5). IEEE.

[2] Dhiman, P., Kaur, A., Hamid, Y., Alabdulkreem, E., Elmannai, H., & Ababneh, N. (2023). Smart Disease Detection System for Citrus Fruits Using Deep Learning with Edge Computing. *Sustainability*, *15*(5), 4576.

[3] Huang, X., Chen, A., Zhou, G., Zhang, X., Wang, J., Peng, N., ... & Jiang, C. (2023). Tomato leaf disease detection system based on FC-SNDPN. *Multimedia Tools and Applications*, *82*(2), 2121-2144.

[4] Chug, A., Bhatia, A., Singh, A. P., & Singh, D. (2023). A novel framework for image-based plant disease detection using hybrid deep learning approach. *Soft Computing*, *27*(18), 13613-13638.

[5] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In *2017 6th International conference on agro-geoinformatics* (pp. 1-5). IEEE.

[6] Ouhami, M., Es-Saady, Y., Hajji, M. E., Hafiane, A., Canals, R., & Yassa, M. E. (2020). Deep transfer learning models for tomato disease detection.

In *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9* (pp. 65-73). Springer International Publishing.

[7] Gonzalez-Huitron, V., León-Borges, J. A., Rodriguez-Mata, A. E., Amabilis-Sosa, L. E., Ramírez-Pereda, B., & Rodriguez, H. (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Computers and Electronics in Agriculture*, *181*, 105951.

[8] Chen, H. C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C. W., Chen, L., & Weng, C. E. (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics*, *11*(6), 951.

[9] Natarajan, V. A., Babitha, M. M., & Kumar, M. S. (2020). Detection of disease in tomato plant using Deep Learning Techniques. *International Journal of Modern Agriculture*, *9*(4), 525-540.

[10] Kaur, J., Agrawal, S., & Vig, R. (2012). A comparative analysis of thresholding and edge detection segmentation techniques. *International journal of computer applications*, *39*(15), 29-34.

[11] Singhal, P., Verma, A., & Garg, A. (2017, January). A study in finding effectiveness of Gaussian blur filter over bilateral filter in natural scenes for graph based image segmentation. In *2017 4th international conference on advanced computing and communication systems (ICACCS)* (pp. 1-6). IEEE.

[12] Ooi, A. Z. H., Embong, Z., Abd Hamid, A. I., Zainon, R., Wang, S. L., Ng, T. F., ... & Ibrahim, H. (2021). Interactive blood vessel segmentation from retinal fundus image based on canny edge detector. *Sensors*, *21*(19), 6380.

[13] Dhiman, P., Kukreja, V., & Kaur, A. (2021, September). Citrus Fruits Classification and Evaluation using Deep Convolution Neural Networks: An Input Layer Resizing Approach. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-4). IEEE.

[14] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

[15] Nayef, B. H., Abdullah, S. N. H. S., Sulaiman, R., & Alyasseri, Z. A. A. (2022). Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks. *Multimedia Tools and Applications*, 1-30.

[16] Seelwal, P., & Sharma, A. (2022, December). Automatic Detection of Rice Diseases using Deep Convolutional Neural Networks with SGD and ADAM. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 1256-1260). IEEE.

[17] Haghighi, S., Jasemi, M., Hessabi, S., & Zolanvari, A. (2018). PyCM: Multiclass confusion matrix library in Python. *Journal of Open Source Software*, *3*(25), 729.

[18] Da Costa, A. Z., Figueroa, H. E., & Fracarolli, J. A. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering*, *190*, 131-144.

[19] Saranya, S. M., Rajalaxmi, R. R., Prabavathi, R., Suganya, T., Mohanapriya, S., & Tamilselvi, T. (2021, February). Deep learning techniques in tomato plant–a review. In *Journal of Physics: Conference Series* (Vol. 1767, No. 1, p. 012010). IOP Publishing.

[20] Mu, Y., Chen, T. S., Ninomiya, S., & Guo, W. (2020). Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors*, *20*(10), 2984.

[21] Rahman, S. U., Alam, F., Ahmad, N., & Arshad, S. (2023). Image processing based system for the detection, identification and treatment of tomato leaf diseases. *Multimedia Tools and Applications*, *82*(6), 9431-9445.