**Research Scholar Praveen GOYAL, Master of Technology (corresponding author)**
**Email: praveen_soit@yahoo.com**
**Jaypee University of Engineering and Technology, India**

**Assistant Professor Dinesh Kumar VERMA, PhD**
**Email: dinesh.verma@juet.ac.in**
**Jaypee University of Engineering and Technology, India**

**Professor Shishir KUMAR, PhD**
**Email: shishir.cs@bbau.ac.in**
**Babasaheb Bhimrao Ambedkar Central University, India**

# DIAGNOSIS OF PLANT LEAF DISEASES USING IMAGE BASED DETECTION AND PREDICTION USING MACHINE LEARNING APPROACH

***Abstract.*** *The increase in the world population consequently prepares the need for food on a fundamental level. As a result, agriculture is important throughout the world. To meet human demand and provide money for farmers, a variety of crops, vegetables, fruits, fish, and animals are raised all year round. Nevertheless, occasionally individuals cultivating crops and grains suffer partial or even entire harm due to a lack of suitable cultivating knowledge, expertise, and awareness of disease prediction. Given the size of the agriculture industry, they are employing artificial intelligence to detect diseases in vegetable plants to make the operation more efficient. To place this concept into practice, this object utilises the basic vegetable "Tomato" to detect leaf illnesses using the deep learning methods of the Convolutional Neural Network (CNN). The tomato output is significantly impacted by tomato leaf disease. The identification of agricultural illnesses is essential for the agricultural industry. Traditional data augmentation techniques like rotation, flipping, and translation have a lot of limitations and cannot produce good generalisation outcomes. Deep learning methods have become effective tools for picture recognition. One of the deep learning technologies, convolutional neural networks, has produced outstanding results in this area. In this study, nine categories of tomato leaf diseases, including one healthy category, were identified. The illness can be predicted by the farmers by entering the signs in the form of photographs of sick tomato leaves. In conclusion, the system displayed an accuracy of over 88%. It is considered to be a user-friendly technology that will assist vegetable farmers, in particular "Tomato" producers, to lessen insect inhibition by identifying leaf diseases and to boost the output by opening up more options for different vegetable diseases investigation and a professional marketplace.*

DOI: 10.24818/18423264/57.4.23.18

***Keywords:*** *convolutional neural network (CNN), data augmentation, data cleansing, Kernel convolution, quintessential learning*

**JEL Classification: Q1, Q18**

## 1. Introduction

The development of modern technologies has enabled the production of enough food to satisfy social needs. The security of the crops and food, nevertheless was not improved. Farmers are in a challenging situation as a result of climate change, a decline in pollinators, plant diseases, and other issues (Baheti et al., 2023). The leaf image ought to be of the highest caliber and easily viewed by everyone. When it involves detecting leaf diseases, the colour of the leaves can differ due to environmental variables, and a variety of diseases can be seen in different types of plants, making identifying illnesses quite difficult. Dataset requirements, noisy data impacting the leaf samples, the process of segmentation that may detect diseases nonetheless samples that need to go through testing and training, and classification are other challenges (Janani and Jebakumar, 2023). Agricultural productivity and cost can be significantly enhanced with the help of a neural network, and the identification of diseases of a plant leaf is possibly exposed to human gloss by visual scan (Narla and Suresh, 2023). Since the plant leaf displays illness signs, it is considered while identifying a disease. Researchers have developed a variety of techniques to improve their ability to recognise different forms of illnesses in leaves (Sahu and Pandey, 2023). In the past, crop diseases were predicted using manual detection techniques. Early automated identification and treatment of diseases in plants remain difficult issues in agriculture (Thakur et al., 2023). This work focuses on creating and implementing a Platform-as-a-Service (PaaS) cloud-based convolutional neural network (CNN)-based real-time predicting diseases system (Zhang et al., 2023). To forecast tea leaf illness, the same dataset is also applied to deep convolutional neural networks (DCNN) like ResNet50, Xception, and NASNetMobile. It classified tomato leaf illnesses and citrus plant diseases using the AlexNet, GoogleNet, and Gabor CapsNet CNN architectures. One of the most produced and consumed veggies in the world is the tomato. (Kukadiya and Meva, 2023).

The control of tomato diseases is a crucial activity that hinders the advancement of crop performance and development and accounts for a sizable portion of overall production costs (Fathima et al., 2023). They discovered that the proposed model performed poorly when utilised to classify images with multiple backgrounds (Haridasan et al., 2023). Because it assesses a variety of factors, including crop output, yield capacity, grain quality, and nutrient retention, diagnosing plant diseases is crucial for determining the quality and health of a crop (Joshi and Bhavsar, 2023). Plant diseases have historically been identified using the more time-consuming and expensive procedures of chemical analysis or visual inspection of leaf symptoms (Sagar et al., 2023). In what way well CNN models work is significantly influenced by the training dataset (Rashid et al., 2023). Despite having outstanding prior model results, the main problem in these studies was that the datasets were obtained from an experimental setting (laboratory), where the conditions for machine perception were comparatively optimal (Anim-Ayeko et al., 2023). With disregard for actual field circumstances. To current knowledge, no research has been done on using CNN models to identify tomato leaf diseases in real-world settings (Seth et al., 2023). The curve let transformation was employed to extract features from an image of a plant leaf, which was then separated into several sub-images before being classified using a multi-class SVM classifier. Improved k-mean clustering is employed in unsupervised segmentation techniques to identify sick plants (Lamba et al., 2023). To identify tomato leaf diseases, two already trained convolutional neural networks were applied to classify images of both good and bad tomato leaves using a dataset gathered in both lab and field settings. The remainder of the section is as follows. Section 2 indicates the literature review, Section 3 explained the current problem definition and motivation of the research. Section 4 exhibited the proposed study techniques. Section 5 denoted experimentation and results discussion. Section 6 ends with the conclusions.

## 2. Literature Review

Pal et al. (2023) proposed an Agriculture Detection (AgriDet) framework that combines traditional Inception-Visual Geometry Group Network (INC-VGGN) and Kohonen-based deep learning networks to identify plant illnesses and categories the severity of damaged plants. The core objective of the AgriDet framework is to aid farmers in obtaining a precise diagnosis of the kind of disease of plants that have developed in the field. In addition to the two new convolutional neural networks (CNN), models suggested by Ulutaş et al. (2023), MobileNetV3Small, EfficientNetV2L, InceptionV3, and MobileNetV2 are employed, which are four other well-known CNN models. The outcomes show that the suggested ensemble models stand out in appreciation of their rapid training and testing times and excellent classification results with an accuracy of 99.60%. Convolutional neural networks (CNNs) were utilised by Lanjewar et al. (2023) to establish and build a real-time disease prediction system on the Platform-as-a-Service (PaaS) cloud. The comparison results show that the CNN model works with amazing accuracy, outperforming the DCNN and literature-reported methods. To extract diseased regions and identify them, Mzoughi and Yahiaoui (2023) suggested a hybrid approach that combines the powers of deep learning-based semantic segmentation with classification skills. The outcomes showed that the disease symptom feature could not surpass the original image, indicating that the dataset's excellent accuracy was also a product of contextual data. According to Javidan et al. (2023) research, black measles, black rot, and leaf blight are three grape leaf diseases that may be identified and classified using a unique image processing method and multi-class support vector machine (SVM). The outcome shows in what way PCA was applied to classify grape leaf output using SVM. The inability to discriminate between diseases at first glance and the similarity between distinct symptoms have become key challenges in the domain, according to Sudhesh et al. (2023) proposal for the task of identifying rice leaf diseases utilising transfer-learned deep learning models. In comparison to other deep features and machine learning techniques, the simulation results show that the DenseNet121 deep feature with the Random Forest classifier works better. Deep learning was proposed by Pal et al. (2023) to classify plant diseases based on images that were directly collected in the field. The findings, assessed using a variety of metrics such as accuracy, recall, precision, and f1-score, demonstrated that the proposed ensemble convolutional neural network surpassed the single convolutional neural network in categorising diseases in real field conditions with variations in brightness, disease similarity, complex background, and multiple leaves.

In a study by Saeed et al. (2023), two pre-trained convolutional neural networks (CNNs) were utilised to distinguish healthy and diseased tomato leaf pictures. The Inception ResNet V2 model with a 15% dropout rate and the Inception V3 model with a 50% dropout rate provided the best performance, with an accuracy of 99.22% and a loss of 0.03 as measured by the most significant results. By enhancing the standard YOLOv5 network using the Feature Enhancement Module (FEM) and Coordinate Attention (CA) approaches, Zhu et al. (2023) investigated an accurate and reliable detection model for apple leaf disease dubbed Apple-Net. The experimental results demonstrate that Apple-Net outperforms four traditional target detection models in terms of mAP@0.5 (95.9%) and precision (93.1%), demonstrating that it is more effective at identifying apple leaf diseases. With the help of additional images captured in the nation's fields, Guerrero-Ibañe et al. (2023) suggested a model based on convolutional neural networks to detect and categorise tomato leaf illnesses using a public dataset. The findings demonstrate that the suggested model performs well in the process of identifying and categorising illnesses in tomato leaves.

## 3. Research Problem Definition and Motivation

Plants are directly or indirectly essential for all other species, including humans. Every ecological system is dominated by plants. It will affect the entire ecological system if there is any damage to the plants. Symptoms on plant leaves can be employed to diagnose plant diseases with the naked eye; nevertheless, the complexity of this method is swiftly growing. Even experienced agricultural experts and botanical pathologists frequently struggle to accurately diagnose specific diseases, which leads them to draw incorrect conclusions and offer unsatisfactory solutions. This failure is frequently caused by the complexity of the situation, the sheer number of cultivated crops, and the phytopathological issues that these crops currently face. The most widely cultivated crop in the world is the tomato. The tomato plants' fruits, stems, roots, and leaflets are all affected by diseases and pests.

On the leaves and leaflets of tomato plants, phonological alterations might take the form of irregular development, discolouration, spots, damages, wilting, desiccation, and necrosis. The widespread adoption of these chemical techniques harms the environment, human health, and plant health. Tomato leaf disease diagnosis: only a small portion of the leaf's surface is affected. When images are captured by field instruments or remote sensing devices, image processing is frequently employed in precision agriculture. However, because the traits of various diseases are similar, it is challenging to distinguish between different types of disease, and disease recognition is inaccurate in a complex natural environment. To automatically extract crucial information about disease features from a complicated environment, this study augments the original CNN network model with an attention module. Plants' leaves are harmed by pests and diseases that need to be found. The actual appearance of the leaf is altered by these adverse impacts, making it possible to identify the source of the harm from photographs obtained by cameras.

## 4. Proposed Research Methodology

The production of tomatoes is substantially impacted by tomato leaf disease in the agricultural industry, and the detection of crop diseases is crucial. The tomato leaf was selected for the suggested study to identify and forecast tomato leaf disease. This study report contains a detailed explanation of the diagnosis and detection procedures for tomato illness. The system was developed and trained using a dataset of tomato leaves and CNN (convolutional neural networks) techniques. An accurate result is to be anticipated as output after processing the provided images.

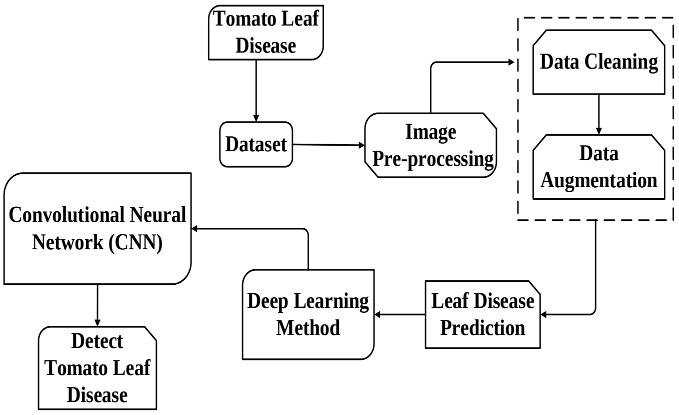

**Figure 1. Process Workflow Diagram**

Figure 1 depicts the suggested method workflow for identifying tomato leaf disease. Pre-processing, feature extraction, and classification are the three stages of the suggested model. Images need first be rebuilt into a format that is acceptable for feature extraction, which is accomplished using image pre-processing techniques. Convolutional neural networks are applied to extract features in the second step, and they are then utilised to classify plant diseases using various types of convolutional neural networks. The result of the optimisation process is the mapping function, i.e., $f\,.A \rightarrow B$, in addition to the training samples. The final premise, $x \approx f$ was derived from all feasible objective functions with various parameters.

### *4.1. Dataset Description*

The plant village open repository dataset has been employed to classify Tomato leaf disease. The diverse dataset of plant leaves, such as the UCI plant leaf dataset, Kaggle, and the Cleared Leaf Image Database, are openly available on the web. Each dataset consists of different plants, leaf images, characteristics, and samples in each class. The Kaggle plant village dataset has been exploited in the proposed study, which contains over 49,000 images of 14 different plants, including strawberry, squash, apple, soybean, corn, grape, potato, tomato, etc.

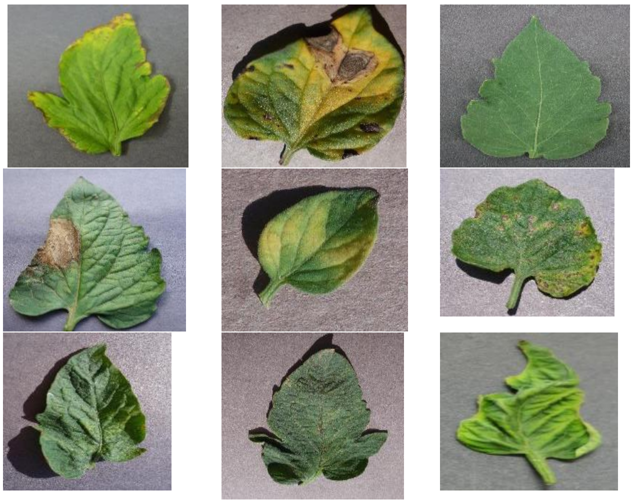

**Figure 2. Sample Images in Dataset**

The tomato leaves were selected as the target crop and the several types of tomato leaves are shown in Figure 2. The CNN model employed in this study was trained using the widely exploited Plant Village Tomato Leaf dataset. The Plant Village dataset, which contains nine classes, was utilised to extract the images of tomato leaves. These classes include Spider mites (5028 images), Bacterial spot (6321 images), Early blight (3000 images), Healthy (4773 images), Late blight (5727 images), Leaf mould (2856 images), Septoria leaf spot (5313 images), Target spot (4212 images), and yellow leaf curl virus 16071 images. The visual quality that the Plant Village dataset offers makes it useful for evaluating plant diseases.

### *4.2. Image Pre-processing*

The primary objectives of image pre-processing are to improve, smoothness, and remove noise brought on by camera flash or high-lighting issues, and increase the effectiveness of the prediction and classification process. A pre-processing stage should be taken into consideration to improve the appearance of the input tomato leaf images and to perform feature extraction. Data cleansing is frequently used to remove artefacts using appropriate data transformation techniques to enhance the modelling learning process. The pre-processed images were enhanced using a variety of techniques, such as clockwise and anticlockwise rotation, horisontal and vertical flipping, zoom intensity rotation, zooming rotation, resizing, zooming, flipping, shifting, brightness, filling rescaling (greyscale), validation split, and shearing. During the instruction process, the images were not duplicated then rather enhanced, and hence the physical copies of the enhanced images were not retained nonetheless rather employed momentarily in the procedure. This augmentation strategy not only stops the model from overfitting and model loss, but also strengthens the model's robustness so that it can more accurately categorise images of real-world plant diseases. When the dataset has a very small number of samples. Since deep learning models cannot function properly in this circumstance, we need augmentation of the data. Therefore, employing transformations like shearing and zooming, the sample of available images may be represented in several ways in such a situation. The dataset of raw images is first divided into 20% verification samples and 80% training samples, and then the data augmentation approach is utilised to change every pixel value to [0-255], using the rescale 1./255 command. The 20% zooming in and out 20% shearing of the image. Flipping the rows or columns of pixels randomly in the horizontal and vertical directions and moving the image along the X and Y axes by 10% each. Increasing the brightness of the image by a factor of [0.25–1.0]. The arbitrarily spinning of the image by a factor of [0–15] changes the black colour's continuous fill mode and, additionally, data splitting to reserve 20% for validation.

### *4.3. Tomato Plant Disease Detection Using Convolutional Neural Network*

In the CNN model, a neuron is connected to a fixed number of neurons in this layer. Consequently, fewer neurons are required to manage smaller weight loads. In a layer of totally connected neurons, each neuron was connected to the others. Convolutional neural networks have four layers. The convolutional layer, the ReLU layer, the pooling layer, and the fully connected layer. In the current study, CNN constructs and classifies a few warped images using input from a tomato plant leaf and performs convolutional operations, and CNN perceives the images.

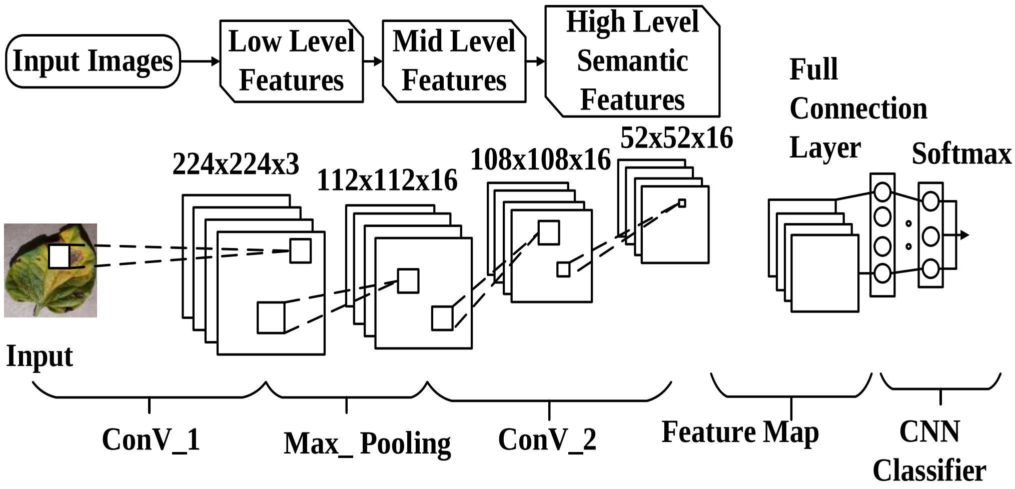

**Figure 3. Proposed CNN Architecture for Tomato Plant Diseases Detection**

The suggested model is constructed with an RGB colour space of 224 x 224 x 3 and is formerly run through eight 5 x 5 filters with a 3 stride. Images are converted to 224 x 224 pixels as the output shape. It employs a second set of sixteen 5 x 5 filters, this time with stride 3, to produce images with an output shape of 224 x 224. By mapping to a 2 x 2 Max pool layer, the size will drop from 224 to 112. There are then two blocks of convolutional and maximum pool layers, each with a filter size of 5 x 5 and a filter of 32 or 64, respectively. The outputs are then flattened into vector form and placed on top of a 512-unit fully connected layer. A 30% dropout was employed, and then a softmax was activated to decrease overfitting. The actual CNN design is exposed in Figure 3, and the results of each layer are shown in Figure 10.

*4.3.1. Convolutional Layer*

The convolutional layer is made up of numerous feature filters that perform the convolutional operation. As a result, the characteristics are now related to two discrete parts of a larger image, and if a match is discovered, the image will be accurately categorised. Before concatenating with each pixel of the image using its pixel characteristic, this layer needs first to align with the filter of the feature in the image. Next, factor in weights to get the addition, which is then divided by the total number of feature pixels. Finally, search for the filtered image. Additionally, the extracted features from the feature map are moved around by the feature filter.

The kernel convolution and key components of CNNs are employed in other computer vision techniques. In this method, the image was altered by utilising the values from a tiny kernel or number matrix of the filter. The parameters of the feature maps can be calculated successfully using the formula provided below. In this case, $u$ stands for the input image and v for the kernel. The initials $t$ and $s$, respectively, stand for the result indexes for the matrix's columns and rows and the computing formula was given to equation (1).

$$G[s,t] = (u * v)[s,t] = \sum_j \sum_k v[j,k]u[s-j,t-k] \tag{1}$$

After applying the filter to the chosen pixel values, each kernel value was multiplied by its corresponding value in the image, and the results were then placed correctly on the output feature map. When performing the valid and same convolution, the image's size will gradually get smaller each time; after a few iterations, the image will be completely undetectable. If the kernel moves across the entire image, the influence on the placed pixels on the periphery will be much smaller than on the centre. Consequently, a portion of the image's information is lost. Padding with an additional border of zeroes is employed to solve both concerns. Because the border around the valid m s input image was almost the same size as the output, the two were of identical size. The width padding for instance will satisfy the equation shown below, where c stands for padding and d for the dimension of the filter (often an odd number). The equation is an example of a calculating equation (2),

$$c = \frac{d-1}{2} \tag{2}$$

The output matrix's shape, taking padding and stride into consideration. Equation (3) contains the stride calculation formula.

$$q_{out} = \left[\frac{q_{in}+2c-f_i}{s} + 1\right] \tag{3}$$

Convolution in stride always involved a one-pixel kernel shift. Nevertheless, step length can also be thought of as convolution layer hyperparameters. The CNN architectural design can select to increase these stages in the pertinent fields as necessary to cover fewer and demand lower spatial dimensions in the feature map.

### *4.3.2. Pooling Layer*

The pooling layer is typically placed following the convolution layer. The convolution layer's output matrix's size is condensed in this layer. Most often, this involves multiplying the pairs of values in three dimensions, using many filters on the same image, performing convolution separately on each of them, using a stack to store the results, and joining the entire stack at the top. The received size of the tensor (3D matrix) satisfies the equation given below, where $q$ is the size of the depiction and $f_i$ is the size filter. Image quantities include the number of channels$q_{ch}$, padding ($c$), stride ($s$), and filters $q_f$. The calculating formula is shown in equation 4,

$$[q,q,q_c] * [f_i,f_i,q_{ch}] = \left[\left\lfloor\frac{q+2c-f_i}{s} + 1\right\rfloor, \left\lfloor\frac{q+2c-f_i}{s} + 1\right\rfloor, q_f\right] \tag{4}$$

The fundamental distinction between highly interconnected neural networks and the approach is that we should employ convolution. There are two steps in forward propagation. The $P$ value, an intermediate value that is calculated in the first step, is what will be employed as the outcome. After providing input data from the previous layer with the $Q$ tensor (including filters), convolution is added $z$ as a bias.

*4.3.3. Activation Layer*

The connection between the input and output layers in neural networks that are artificial is provided by the activation function as a curvilinear relationship, which additionally affects network performance. The activation function allows for non-linear learning of the network. The nonlinear ReLU (Rectified Linear Unit) activation function is typically utilised in CNN, yet, other activation functions, including linear, sigmoid, and hyperbolic tangent, exist. Values that are less than zero in ReLU are converted to zero, whereas values that are more than zero remain the same nonetheless with (1) applied.

$$f(x) = \left\{\begin{matrix} 0, & if x < 0 \\ x, & otherwise \end{matrix}\right\} \tag{5}$$

The application of the non-linear function of activation $g$ to the intermediate value comes next. The calculating formula is shown in equation (6).

$$P^{[L]} = Q^{[L]} \cdot R^{[L-1]} + Z^{[I]}, Where\ R^{[L]} = g^{[L]}\left(P^{[L]}\right) \tag{6}$$

If the inputs are less than 0, the output result is also 0, as shown in Figure 5(a), of the corrected liner unit (ReLU layer), which is utilised simply to activate the node. Additionally, it exhibits a linear type relationship with a dependent variable if the input is greater than zero. In the event of a negative result, remove it, and then have this method convert it to 0. To produce the feature map, which was then corrected as an output, this was then applied to all of the image's features. Another function for activation sigmoid has also been employed; it is depicted as an S-shaped curve in Figure 5(b). It is typically applied when performing calculations that the anticipated outcome should be interpreted as likelihood. The output value is between [0, 1] in the range.

Maximum results were returned for each square segment of size $Zq \times Zq$ in each activation map. Max-pooling was the term for this technique. However, if a stride of 1 was employed, it would produce a novel layer with the dimensions $(Wq - Zq + 1) \times (Cq - Zq + 1) \times hq$. However, we typically utilise strideOq>1 while pooling. Thus, the new layer's length will now be $(Wq - Zq)/Oq + 1$ plus its breadth is $(Cq - Zq)/Oq + 1$. The spatial size map of each activation is thereby greatly reduced by pooling. At each map activation level during convolution processes, pooling is carried out. Convolution works by simultaneously using each high-definition map feature along with a filter to produce a single feature value, whereas pooling works separately on each map feature to produce additional map features. The final result of flattening the feature map of the pooled data was to feed it to a fully connected layer where genuine classification predicts and produces the output. Fully connected networks employed the soft maximum activation function, a sort of logistic regression that follows the probability distribution and normalises the input value to the values vector [0, 1]. As a result, it is referred to as a maximum entropy classifier.

$$g\left(d = i/\theta^{(r)}\right) = \frac{e^{\theta^{(r)}}}{\sum_{j=0}^{p} e^{p\theta^{(r)}}}, Where\ \theta = E_0F_0 + E_1F_1 + \cdots\ E_pF_p \tag{7}$$

In machine learning, regularisation solved the overfitting problem by adding some penalty with a loss function. Those models that custom the dropout layer provide a very numerically less expensive and astonishingly effective regularisation method to minimise overfitting issues while improving the error of generalisation.

### *4.3.4. Fully Connected Layer*

The convolution, pooling, and activation procedures are finished, and the final generated matrix is provided as input into the fully connected layer. This layer carries out recognition and classification. All of the nodes in the layer above and below it were close to one another. There should be the required number of nodes accessible for the fully connected layer. Dense nodes are another term for the fully connected layer. Each layer was connected to all preceding and succeeding levels utilising the edges of the neurons. Utilising connectivity has become essential for any information exchange. All communications are completed when the edge utilised for communication between neurons is removed using regularisation.

The suggested CNN model also calculates the fundamental learning rate. The learning rate is one of the most important hyperparameters for fine-tuning deep network training. The weights are updated while the network is being trained. Between 0.0 and 1.0 is the range for the learning rate value. The learning rate also affects the speed at which the model adjusts to the job. Lower learning rates require more training epochs, whereas higher learning rates require fewer training epochs because of the lower modifications to the weights of each update.  Since the learning rate needs to be properly selected, deep learning networks are challenging to train. It can be the most important hyperparameter for the model. Because initial arbitrary weights are far from optimal, training should start with a reasonably high learning rate. Later in training, the learning rate may decrease to allow for more precise weight updates.

## **5. Experimentation and Result Discussion**

The Kaggle plant village dataset, which includes 26856 digital images of tomato plant leaves from 9 different classes, was employed in this study to test the suggested approach. The suggested system similarly employs a cross-validation ratio, using 70% (21485 images) of the dataset from 9 distinct categories for training and the remaining 30% (5371 images) for validation. The dataset is also exercised to train the model, which helps the algorithm learn in what way to recognise plants in images. An objective evaluation of an algorithm that was successfully fitted to training data is provided by the validation.

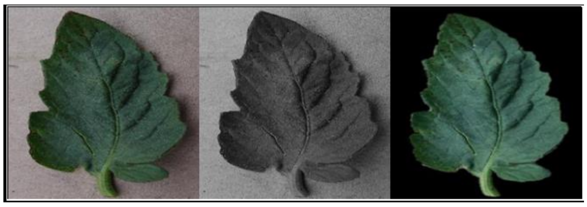

**Figure 4. Augmented Leaf Disease Images (a) Colour (b) Grayscale (c) Segmented**

The data cleaning and augmentation process is finished, and additional training examples were produced by applying a range of transformation techniques to the images. Figure 4 shows the augmented leaf colour will change from the process of pre-processing. The greyscale denotes the scale of the affected leaf and the leaf segmented for the process of data augmentation.

**Table 1. Augmentation of Data**

| S. No. | Transformation Operations | Properties |
|---|---|---|
| 1 | Range of Rotation | 15 degrees clockwise and anti-clockwise |
| 2 | Width Shift Range | 1/10 portion of the whole width |
| 3 | Height Shift Range | 1/10 portion of the whole height |
| 4 | Zoom Range | 0.2% smaller or larger than the original image |
| 5 | Validation Split | 0.2 or 20% of the data was taken as a validation dataset |
| 6 | Rescale | Values to normalize input multiplied 0.0039 with image channel |
| 7 | Resize Image | 224×224×3 pixels |

In this study, data augmentation is recommended for the tomato dataset. Table 1 divides the transformation processes and attributes based on the source dataset. The tomato leaf preprocessing is 15 degrees, the width and height shift range is 1/10 of that, leaf zooming is 0.2%, the image validation split is 0.2 or 20%, and the image resize pixels are 224 by 224 by 3 pixels. All of these operations are applied to supplement the data.

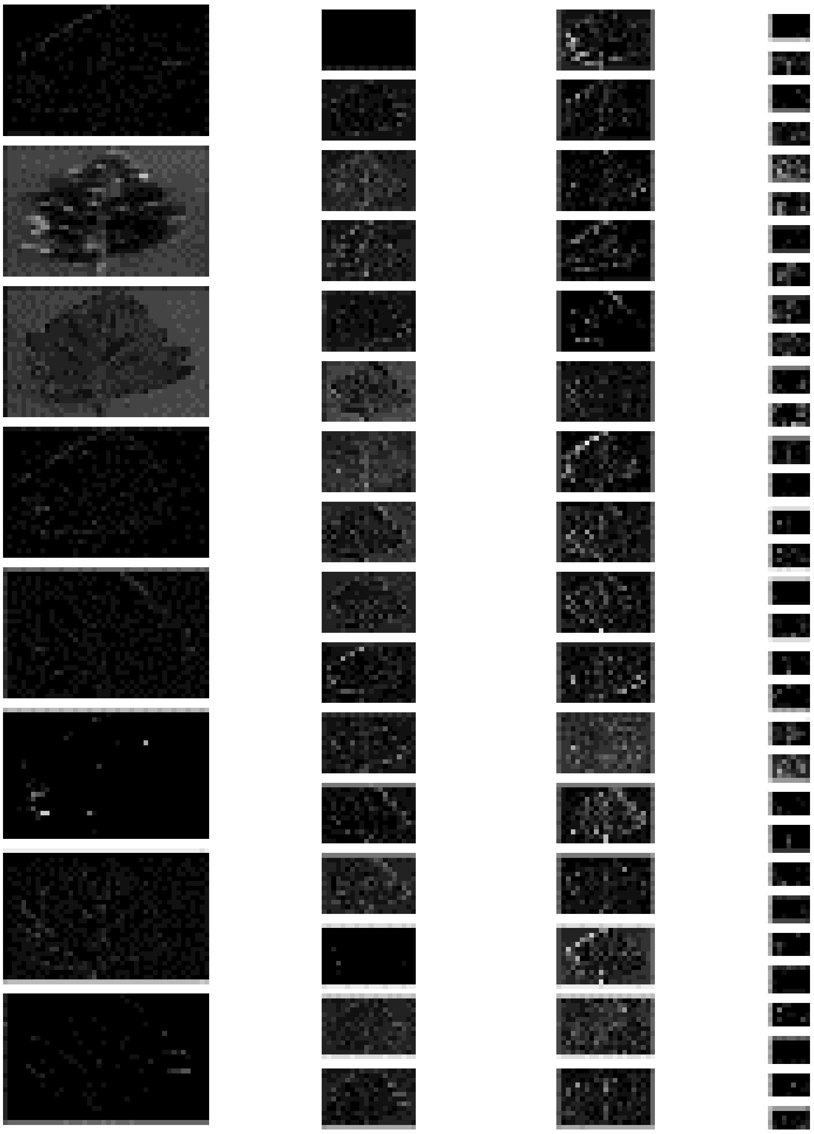

**Figure 5. Layer-by-Layer Image Outputs**

The intermediate output values of the sample layers were shown in Figure 5 along with a tomato leaf that exhibited early blight. The names of the layer or inception module and the output array widths are listed after each intermediate output. Layer by layer, the leaves' basic structure is revealed. The results are shown in this section after each layer of the photographs has been checked for disease.

**Table 2. Summary of the Model (Model "Sequential")**

| S. No. | Types of layers | Shape of outputs | Feature map | Size of filter | Receptive Field | Coverage (%) | Capacity (%) |
|---|---|---|---|---|---|---|---|
| 1 | conv2d (Conv2D) | (None, 224, 224, 8) | 608 | 5×5 | 9 | 2.44 | 100 |
| 2 | conv2d_1 (Conv2D) | (None, 224, 224, 16) | 3216 | 5×5 | 22 | 3.45 | 34 |
| 3 | max_pooling2d (MaxPooling2D) | (None, 112, 112, 16) | 0 | 5×2 | 31 | 4.88 | 41 |
| 4 | conv2d_2 (Conv2D) | (None, 108, 108, 16) | 6416 | 5×5 | 46 | 8.45 | 8.33 |
| 5 | conv2d_3 (Conv2D) | (None, 104, 104, 16) | 6416 | 5×5 | 54 | 9.67 | 9.84 |
| 6 | max_pooling2d_1 (MaxPooling2 | (None, 52, 52, 16) | 0 | 5×5 | 63 | 15.53 | 3.67 |
| 7 | conv2d_4 (Conv2D) | (None, 48, 48, 32) | 12832 | 5×5 | 71 | 21.78 | 3.97 |
| 8 | conv2d_5 (Conv2D) | (None, 44, 44, 32) | 25632 | 5×5 | 108 | 33.47 | 4.34 |
| 9 | max_pooling2d_2 (MaxPooling2 | (None, 22, 22, 32) | 0 | 5×5 | 122 | 37.98 | 2.89 |
| 10 | conv2d_6 (Conv2D) | (None, 6, 6, 64) | 51264 | 5×5 | 188 | 41.96 | 1.78 |
| 11 | flatten (Flatten) | (None, 2304) | 0 | 5×5 | | | |
| 12 | dense (Dense) | (None, 512) | 1180160 | 5×5 | | | |
| 13 | dropout (Dropout) | (None, 512) | 0 | 5×5 | | | |
| 14 | dense_1 (Dense) | (None, 9) | 4617 | 5×5 | | | |
| | **Total Parameters.** | 1,291,161 | | | | | |
| | **Trainable Parameters.** | 1,291,161 | | | | | |
| | **Non-trainable Parameters.** | 0 | | | | | |

CNNs typically achieve good performance with relatively small model sizes because of their ability to learn characteristics directly from the input data and their custom of shared weights in the convolutional layers. In comparison to models with more parameters, such as fully linked deep neural networks, they may be simpler to train and take up less storage space. The objective of this part is to generate new CNN models that require less training time and storage space than the previous models mentioned in the study. Table 3 provides a list of the exact values for each layer in our proposed model. The input and output forms for each layer are also shown. The output shape is computed differently for the convolution and pooling layers. The table lists the different layer types, their output shapes, feature maps, filter sizes, receptive fields, coverage areas, and capacities. Conv2D layers are capable of 100% of their maximum capacity. The total number of parameters that influence the outputs are shaped is 1,291,161, including 1,291,161 trainable parameters.

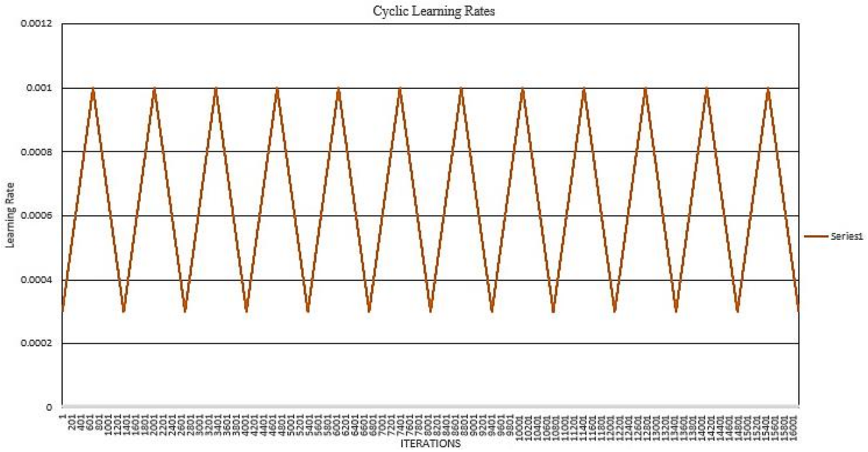

**Figure 6. Triangular Cyclic Learning Rate for Each Iteration**

The Cyclical Learning Rate (CLR), as shown in Figure 6, is a technique for calculating global learning rates for training neural networks that eliminate the need to conduct numerous tests to find the ideal values. This graphic explains the learning rate plot, which cyclically begins at our lower learning rate bound, grows to the highest point at half a cycle, and then lowers again to the lower bound, completing the cycle.

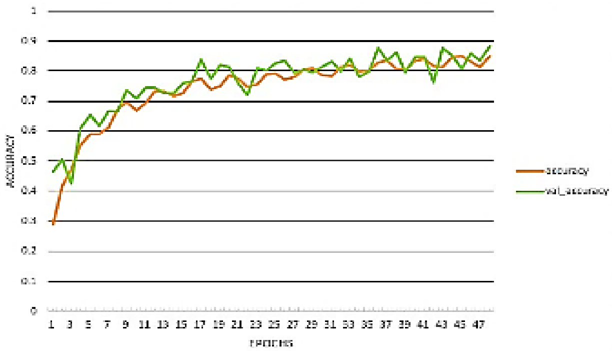

**Figure 7. Accuracy vs Epoch**

Figure 7 shows the loss and precision of the model according to each training and validation set epoch. Meanwhile, the epochs and the regularisation carried out using the dropout approach, the model's accuracy fluctuates. The accuracy and valuation accuracy levels are shown in Figure 8. If the training dataset of the model overperforms or underperforms, overfitting and underfitting may have detrimental effects on the accuracy graph. It made sure the accuracy graph produced the greatest possible, highest accuracy by preventing any overfitting or underfitting.

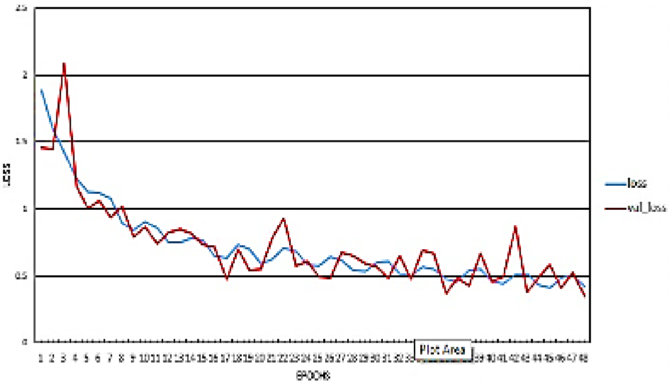

**Figure 8. Loss vs. Epoch**

Accuracy improves as the number of epochs increases and the loss percentage falls. After 48 iterations, the recommended model achieves an accuracy of 88 % in classification without overfitting. Following that, the loss gap between both the training and validation loss curves begins to expand and both curves begin to flatten.

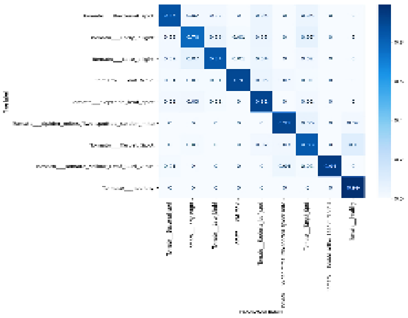

**Figure 9. Confusion Matrix for the Prediction of 8 Tomato Diseases**

The confusion matrix discovered during the assessment of the suggested model is displayed in Figure 9. The true positive (TP), True Negative (TN), False Positive (FP), and false negative (FN) values attained for each class assessed are displayed in the confusion matrix. It displays the confusion matrix for the prediction of nine tomato diseases, with the proper categorisations for each class displayed diagonally in Figure 9. It displays the proportion of nine tomato illnesses that are incorrectly categorised. Additionally, it shows 88 % accuracy in classifying the majority of disorders.

**Table 3. Confusion Matrix Report**

| Name of Disease | Precision | Recall | F1 - Score | Support |
|---|---|---|---|---|
| Bacterial Spot | 0.93 | 0.86 | 0.89 | 600 |
| Early Blight | 0.84 | 0.78 | 0.81 | 600 |
| Late Blight | 0.89 | 0.83 | 0.86 | 600 |
| Leaf Mould | 0.97 | 0.90 | 0.93 | 571 |
| Septoria Leaf Spot | 0.80 | 0.91 | 0.85 | 600 |
| Spider Mite | 0.90 | 0.91 | 0.91 | 600 |
| Target Spot | 0.78 | 0.83 | 0.80 | 600 |
| Leaf Curl Virus | 0.99 | 0.94 | 0.96 | 600 |
| Health | 0.88 | 0.99 | 0.93 | 600 |
| **Accuracy** | | | **0.88** | **5371** |
| **Macro Average** | **0.89** | **0.88** | **0.88** | **5371** |
| **Weighted Average** | **0.89** | **0.88** | **0.88** | **5371** |

Table 3 displays the findings from calculations made for each class of tomato disease leaves for precision, recall, and f1-score. The confusion matrix report demonstrates that the accuracy, precision, recall, F1 score, and support macro averages for tomato leaf disease are identical at 0.89%, 0.88%, and 5371 and that they are also equal when weighted at 0.89%, 0.88%, and 5371. Figure 13 displays the Confusion Matrix for the Prediction of 8 Tomato Diseases, which is employed to gauge the degree of reparability among classes.

### 6. Research Conclusions

The detection of illnesses from images of plant leaves is one of the most important areas of research in precision agriculture. The discipline of precise plant care and growth has the potential to be expanded and advanced by breakthroughs in artificial intelligence, image processing, and graphical processing units (GPUs). Learning models require strong observational abilities to distinguish the various signs of every sickness. Many different visible signs are produced by most plant diseases. The tomato is an essential crop with important culinary and medicinal exploits. Large-scale illnesses are more likely to be encountered throughout the tomato growing process the larger the planting area, which can easily result in significant financial losses. Most tomato illnesses manifest as lesions in the leaves, which can exhibit a variety of symptoms, in the early stages of the disease. Using data from the Kaggle Plant Village dataset, a deep learning-based identification method for tomato plant diseases has been developed in this study. With state-of-the-art performances, a deep neural network can classify and identify images of tomato leaves. The suggested model has been trained and validated using raw image data in a 70.30 ratio. The broadcast can properly classify unknown leaf illnesses by using plant images to learn the characteristics of provided leaf diseases. As a result, the accuracy of classification varied from 12% to 99% and reached an average accuracy of 88%. Although the formed method may provide a fast and simple method for correctly identifying plant leaf diseases, it is not intended to replace the work of human taxonomists. Research in fields such as biotechnology and genetics is being affected by the shortage of taxonomists or individuals who can accurately identify plant samples. By incorporating additional plant leaf datasets, the proposed method may be utilised in the future to enhance the segmentation and identification of diseased leaves from foliage images.

## REFERENCES

[1] **Anim-Ayeko, A.O., Schillaci, C., Lipani, A. (2023),** ***Automatic Blight Disease Detection in Potato (Solanum tuberosum L.) and Tomato (Solanum lycopersicum, L. 1753) Plants using Deep Learning***. *Smart Agricultural Technology*, 100178;

[2] **Baheti, H., Anuradha, T., Yash, B., Sudarshan, D., Om, D. (2023),** ***Tomato Plant Leaf Disease Detection Using Inception V3.*** *In Intelligent Systems and Applications. Select Proceedings of ICISA 2022, Singapore, Springer Nature Singapore*, 49-60;

[3] **Fathima, S.A., Nandhini, M. (2023),** ***Performance Comparison of the Classifiers for Betel Vine Disease Prediction.*** *In Machine Learning, Image Processing, Network Security and Data Sciences. Select Proceedings of 3rd International Conference on MIND 2021, Singapore, Springer Nature Singapore*, 229-245;

[4] **Guerrero-Ibañez, A., Reyes-Muñoz, A. (2023),** ***Monitoring Tomato Leaf Disease through Convolutional Neural Networks***. *Electronics*, 12(1), 229;

[5] **Haridasan, A., Thomas, J., Raj, E.D. (2023),** ***Deep learning system for paddy plant disease detection and classification***. *Environmental Monitoring and Assessment*, 195(1), 120;

[6] **Janani, M., Jebakumar, R. (2023),** ***Detection and classification of groundnut leaf nutrient level extraction in RGB images***. *Advances in Engineering Software* 175, 103320;

[7] **Javidan, S.M., Ahmad, B., Keyvan, A.V., Ampatzidis, Y. (2023),** ***Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning***. *Smart Agricultural Technology*, 3, 100081;

[8] **Joshi, B.M., Bhavsar, H. (2023),** ***Deep Learning Technology based Night-CNN for Nightshade Crop Leaf Disease Detection*****.** *International Journal of Intelligent Systems and Applications in Engineering*, 11(1), 215-227;

[9] **Kukadiya, H., Meva, D. (2023, January),** ***Automatic Cotton Leaf Disease Classification and Detection by Convolutional Neural Network.*** *In Advancements in Smart Computing and Information Security. First International Conference, ASCIS 2022, Rajkot, India, November 24–26, 2022, Revised Selected Papers, Part I Cham, Springer Nature Switzerland,* 247-266;

[10] **Lamba, S., Vinay, K., Anupam, B., Shalli, R. Ahmed, S.H. (2023),** ***A Novel Hybrid Severity Prediction Model for Blast Paddy Disease Using Machine Learning.*** *Sustainability,* 15(2), 1502;

[11] **Lanjewar, M.G., Panchbhai, K.G. (2023),** ***Convolutional neural network-based tea leaf disease prediction system on smartphone using paas cloud.*** *Neural Computing and Applications,* 35(3), 2755-2771;

[12] **Mzoughi, O., Yahiaoui, I. (2023),** ***Deep learning-based segmentation for disease identification***. *Ecological Informatics*, 102000

[13] **Narla, V.L., Suresh, G. (2023, January),** ***Multiple Feature-Based Tomato Plant Leaf Disease Classification Using SVM Classifier*****.** *In Machine Learning, Image Processing, Network Security and Data Sciences. Select Proceedings of 3rd International Conference on MIND 2021 Singapore, Springer Nature Singapore,* 443-455;

[14] **Pal, A., Kumar, V. (2023),** ***AgriDet. Plant Leaf Disease severity classification using agriculture detection framework***. *Engineering Applications of Artificial Intelligence,* 119, 105754.

[15] **Rashid, J., Khan, I., Ali, G., Ur Rehman, S., Alturise, F., Alkhalifah, T. (2023),** ***Real-Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique.*** *CMC-COMPUTERS MATERIALS & CONTINUA*, 74(1), 1235-1257;

[16] **Saeed, A., Abdel-Aziz, A.A., Mossad, A., Abdelhamid, M.A., Alkhaled, A.Y., Mayhoub, M. (2023),** ***Smart Detection of Tomato Leaf Diseases Using Transfer Learning-Based Convolutional Neural Networks***. *Agriculture,* 13(1), 139;

[17] **Sagar, S., Singh, J. (2023),** ***An experimental study of tomato viral leaf diseases detection using machine learning classification techniques***. *Bulletin of Electrical Engineering and Informatics,* 12(1), 451-461;

[18] **Sahu, S.K., Pandey, M. (2023),** ***An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model***. *Expert Systems with Applications*, 214, 118989;

[19] **Seth, V., Paulus, R., Kumar, A. (2023),** ***Tomato Leaf Diseases Detection Using Deep Learning—A Review.*** *Intelligent Systems and Smart Infrastructure. Proceedings of ICISSI 2022*, 118;

[20] **Sudhesh, K.M., Sowmya, V., Sainamole, K., Sikha, O.K. (2023),** ***AI-based rice leaf disease identification enhanced by Dynamic Mode Decomposition.*** *Engineering Applications of Artificial Intelligence,* 120, 105836;

[21] **Thakur, P.S., Sheorey, T., Ojha, A. (2023),** ***VGG-ICNN. A Lightweight CNN model for crop disease identification.*** *Multimedia Tools and Applications,* 82(1), 497-520;

[22] **Ulutaş, H., Aslantaş, V. (2023),** ***Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model***. *Electronics*, 12(4), 827;

[23] **Zhang, Y., Huang, S., Zhou, G., Hu, Y., Li, L. (2023),** ***Identification of tomato leaf diseases based on multi-channel automatic orientation recurrent attention network.*** *Computers and Electronics in Agriculture*, 205, 107605;

[24] **Zhu, R., Zou, H., Li, Z., Ni, R. (2023),** ***Apple-Net. A Model Based on Improved YOLOv5 to Detect the Apple Leaf Diseases***. *Plants,* 12(1), 169.