

Next-Generation Plant Disease Detection: A Efficient Approach to Plant Disease Identification with HW-CNNs and Wasserstein Metrics

Daya Shankar Verma

Indian Institute of Information Technology Ranchi

Jitendra K. Mishra

Indian Institute of Information Technology Ranchi

Ankit Kumar

Guru Ghasidas Vishwavidyalaya

Linesh Raja

`linesh.raja@jaipur.manipal.edu`

Manipal University Jaipur

Vimal Dungdung

Banaras Hindu University

Achintya Singhal

Banaras Hindu University

Article

Keywords: Plant leaves, Diseases, SVM, KNN, Deep learning, CNN

Posted Date: June 25th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-6904746/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Next-Generation Plant Disease Detection: A Efficient Approach to Plant Disease Identification with HW-CNNs and Wasserstein Metrics

Daya Shankar Verma¹, Jitendra K. Mishra², Ankit Kumar³, Linesh Raja⁴, Vimal Dungdung⁵, Achintya Singhal⁶

¹ Department of Computer Science and Engineering, Indian Institute of Information Technology Ranchi, Jharkhand, 835217, India; dsverma.rs@iiitranchi.ac.in

² Department of Electronics and Communication Engineering, Indian Institute of Information Technology Ranchi, Jharkhand 835217, India; jkmishra@iiitranchi.ac.in

³ Department of Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, India; ankitkomar@ieee.org

⁴ Department of Computer Applications, Manipal University Jaipur, Jaipur, Rajasthan, India, linesh.raja@jaipur.manipal.edu

^{5,6} Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi, vimaldungdung@bhu.ac.in, achintya@bhu.ac.in

* Corresponding author. linesh.raja@jaipur.manipal.edu

Abstract: Timely identification and treatment of plant diseases are essential for boosting agricultural productivity and reducing economic losses. In this study, we present an innovative deep learning framework for patient-automated plant disease detection using a Hierarchical Wasserstein Convolutional Neural Network (HW-CNN). Specifically, we introduce depth-separable convolutions for computational cost savings and a new Hierarchical Wasserstein Distance (HWD) loss function which improves classification by leveraging inter-class relationships. The model was trained and validated on a large dataset containing 53200 Images across 38 different diseases in 14 different species of plants. Additionally, the proposed methodology provides a detailed description of the preprocessing steps (transformation of colour space to H, S, and V, pixel masking of green pixels), feature extraction using Hu moments, Haralick texture features, and colour histograms. The HW-CNN architecture is based on depth-separable convolutions which have been shown to yield very good performance with fewer parameters. The HWD loss function also helps build a more suitable loss landscape that enables the model to generalise across different types of diseases. The HW-CNN outperformed classical machine learning models (SVM, Random Forest, and Logistic Regression) and other deep learning architectures with an accuracy of 99.19%. The experimental results show that the HW-CNN has an accuracy of 99.19%. The experimental results showed similar improvements in performance, while significantly reducing complexity compared to existing methods. Throwing light on the effectiveness of advanced deep learning techniques to overcome significant obstacles in plant disease detection, including serendipitous symptoms and climate differences. The novel HW-CNN architecture forms a scalable, low-power circuitry with high energy efficiency that can tremendously benefit real-world scenario applications such as agriculture, reduce potential crop losses, and improve food security in society.

Keywords: Plant leaves, Diseases, SVM, KNN, Deep learning, CNN

1. Introduction

Improved agricultural yields and fewer monetary losses can result from the faster identification and treatment of sick plants, both of which are facilitated by automated plant disease detection. Automatic illness identification using machine learning and computer vision techniques has made significant strides in recent years. To diagnose plant diseases, these techniques employ deep learning models and image-processing algorithms. Several machine learning techniques other than CNNs have been tried for plant disease diagnosis, including support vector machines (SVMs) and decision trees. These

technologies combine image processing with statistical analysis to identify telltale signs of illness. The agricultural sector in India and elsewhere may benefit greatly from automated disease identification using machine learning and computer vision techniques. Farmers can increase crop yields, decrease economic losses, and guarantee food security to millions of people if they can swiftly detect and treat sick plants. The time and energy spent on crop monitoring and disease identification can be drastically reduced by automating the detection of plant diseases using machine learning and computer vision techniques. Images of plant leaves recorded by digital cameras and sensors can then be analysed using image processing algorithms and deep learning models to detect the presence of illness. This method has the potential to be very useful in finding infected plants, even in extensive plantings, and in providing farmers with the knowledge they need to take preventative steps. To avoid wasting time and energy on human monitoring and inspection, farmers may benefit from using automated detection systems. In addition, automated detection technologies can assist in determining the nature of the plant disease, which is useful in deciding how to treat it. This can help save costs by preventing the illness from spreading to other plants in the crop.

The use of an image retrieval strategy in conjunction with machine learning models to identify plant diseases is an interesting idea. The program can reliably determine if a leaf is healthy or diseased by evaluating the observed patterns of illnesses on plant leaves and applying image processing techniques. Classifying plant leaves as either healthy or unhealthy is a typical use of machine learning models such as SVM, Random Forest, Naive Bayes, Logistic Regression, and Convolutional Neural Networks (CNN). These algorithms can analyse a large database of labelled plant images to learn how to detect diseases. The lack of specialised tools or lengthy training required to use this method is one of its many benefits. You may obtain a rapid analysis of the plant's condition by taking pictures of its leaves using your phone or digital camera and uploading them to the model. The model can precisely determine plant illnesses by employing picture retrieval methods. The knowledge gained from this study will aid farmers in making informed decisions about disease treatment and containment of the epidemic. This determination is made using a machine learning method and is performed in stages across several leaves. For classification, as illustrated in Figure 1, we use SVM, Random forest, Naive Bayes, Logistic regression, and convolution neural networks.

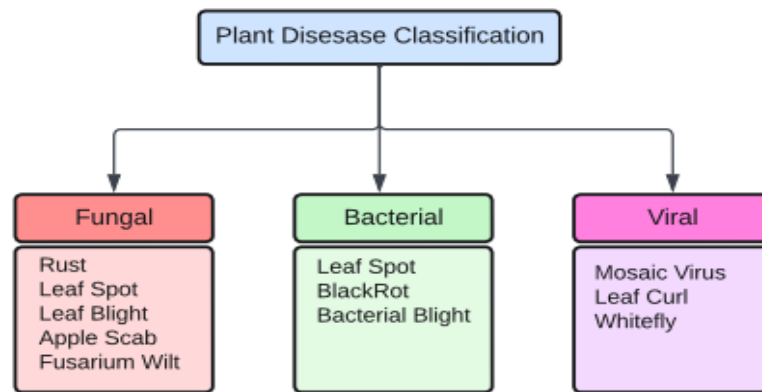


Figure 1: The categorization of plant diseases

1.1. Disease Categories

Plant diseases can be broadly categorised into three types: Fungal, Bacterial, and Viral.

Fungal: A plant's leaves, stems, and roots are susceptible to fungal infections. Powdery mildew, downy mildew, rust, and anthracnose are just a few examples of fungal diseases that may harm plants. Yellowing, wilting, browning, spotting, and staining of leaves are some of the indications of these illnesses, along with slowed development and decreased production. Fungal illnesses are notoriously difficult to identify and contain because of their rapid spread and indistinguishability from other plant ailments. Diseases caused by fungi are notoriously difficult to diagnose, but modern image processing and machine learning methods have made this a reality. Downy mildew appears as yellowish-green

blotches on the underside of leaves, whereas powdery mildew causes white powdery growth on the surface of leaves. Nevertheless, leaf spots of a reddish-brown colour are caused by rust, whereas dark, deep patches are caused by anthracnose. The model may learn to detect these patterns and effectively diagnose fungal illnesses in plants by employing machine learning models such as convolutional neural networks (CNNs). This information may be used to assist farmers to stop the spread of the illness and save their crops before they are completely ruined.

Bacteria: Many different types of bacteria may infect plants, and these bacteria produce illnesses known as bacterial diseases. Bacterial infections of plants cause wilting, leaf spots, necrosis, and blights. Leaf blight, fire blight, and canker are examples of bacterial plant diseases. These pathogens can severely impair agricultural production, lowering both output and quality. Several different symptoms, including wilting, rotting, and leaf spots, can be caused by bacterial plant diseases. Many different types of plant diseases can be caused by bacteria, including bacterial blight, canker, and wilt. Several of these diseases can have devastating effects on crop production and quality.

Viral: Plant viruses can infect many different parts of a plant, from the leaves to stems to fruits. The development of a plant may be impeded, its leaves may be discoloured or mottled, and its structures may be warped if it is infected with a virus. Cucumber mosaic virus, tomato spotted wilt virus, and other mosaic viruses are examples of plant diseases caused by viruses. When a plant is infected with a virus, the effects can be devastating on its development and productivity, and there are typically no treatments available. Therefore, the most effective methods for controlling viral plant diseases are prevention and early identification. Visual signs, such as deformed or discoloured leaves, stunted development, and strange patterns or markings on the leaves, are often used to diagnose viral infections.

1.1.1. Fungal Disease

Fungal diseases are caused by pathogenic fungi and can have a wide range of harmful effects on plants. Because of the severity of their effects, fungal infections pose a danger to agricultural productivity worldwide. The use of fungicides, culture methods, and resistant crop cultivars are all useful management options. The development of automated detection methods based on machine learning and image processing technologies can be invaluable tools for the early identification and diagnosis of fungal illnesses, which are vital for their successful control. Rust, apple scab, Fusarium wilt, leaf spot, and leaf blight are some of the most prevalent fungal diseases. Rust is caused by various fungal species such as *Phragmidium* spp., *Gymnosporangium juniperi-virginianae*, and *Puccinia sorghi*, which affect different plants such as roses, apples, and corn. Apple scab is caused by the fungus *Venturia inaequalis* [2] and causes pale yellow or olive green spots on leaves. Fusarium wilt is caused by the pathogen *Fusarium oxysporum* and its various forms, such as *Fusarium oxysporum* f. sp. *vasinfectum* which causes cotton fusarium wilt. Leaf blight is caused by different fungi, such as *Helminthosporium turcicum*, *Alternaria*, *Phytophthora infestans*, and *Setosphaeria turcica*. Black rot is also caused by different fungi such as *Diplodia seriata* and *Guignardia bidwellii* which affect different plants such as apples and grapes. These diseases can cause significant damage to crops and significantly affect crop yields if not properly controlled and managed, as shown in Figure 2.



Apple Scab



Rust



Fursairum Wilt



Leaf Blight



Leaf Spot

Figure 2: Fungal disease categories

1.1.1. Bacterial Disease

Bacterial canker, like bacterial wilt, is caused by bacteria such as *Pseudomonas syringae* and manifests as sunken, necrotic lesions on plant stems, leaves, and fruits. *R. solanacearum* is responsible for bacterial wilt, which causes the leaves to wilt and turn yellow before ultimately killing the plant. Without effective management, these bacterial infections can spread rapidly and destroy crops. Disease-resistant cultivars, crop rotation, good sanitation, and chemical treatments such as antibiotics and copper-based compounds are all effective ways to combat bacterial infections. Bacterial leaf spot is caused by various types of bacteria, including *Pseudomonas* and *Xanthomonas*. It is more prevalent in warm, humid conditions and can cause symptoms such as reddish-brown circular specks on the leaves. Bacterial leaf blight caused by *Xanthomonas citri* pv. *malvacearum* can result in severe yellowing, browning, spotting, wilting, or death of plant tissues (Figure 3).



Blackrot



bacterial blight



leaf Spot

Figure 3: Bacterial disease categories

1.1.2. Viral Disease

Plants as diverse as cauliflower, cotton, tomatoes, and cucumbers are susceptible to viral illnesses, the result of viral infections. Leaf curl, whitefly, and mosaic virus are the three most prevalent viral illnesses. Curling, puckering, yellowing, and shortening of veins are all symptoms of leaf curl, which affects plants ranging from chilies and cotton to tomatoes. The whitefly *Bemisia tabaci* is responsible for the Cotton Leaf curl virus, which causes the tiny and big veins in the leaves to thicken and the leaves to curl both upwards and downwards. DNA viruses in the genus *Begomovirus* cause leaf-curling symptoms in plants, and two of them are the Chilly Leaf Curl Virus and the Tomato Leaf Curl Virus. As shown in Figure 4, the Tomato Mosaic Virus (TMV), which belongs to the *Tobamoviridae* family and *Tobamovirus* genus, causes patches of yellowing and deeper greening on the leaves of infected plants.



Leaf Curl



Whitefly



Mosaic Virus

Figure 4: Viral disease categories

1.2. Challenge identified

Scientists and researchers have studied plant leaf diseases to better understand the issues that need to be addressed [4]. Several potential examples are presented below. The representation of the leaf must be precise.

- I. The public needs unrestricted access to the datasets.
- II. There was some background noise in the data gathered from the leaves.
- III. Diseases can be more easily diagnosed using segmentation, but only after the samples have undergone extensive

testing and training.

- IV. In certain circumstances, leaf colour may vary owing to environmental conditions.
- V. A major obstacle to early illness identification is the wide variety of plant species that may be infected by a given pathogen.

1.3. Major Contribution:

- i. Review the existing work done by various researchers and highlight the experience in training models to correctly classify plant diseases based on visual symptoms visible on leaves. Furthermore, it pointed out weaknesses in current approaches, including deficiencies in feature extraction and classification performance, particularly in distinguishing maladies with indistinguishable symptoms or slight differences.
- ii. Different plant health and disease detection approaches have been investigated, including traditional methods (manual inspection, molecular diagnostics) as well as artificial intelligence (AI), incorporating machine learning (ML) and deep learning (DL)-based methods. The analysis highlighted the pros and cons of these techniques, including the highly computational nature of ML/DL models or the requirement for large labelled datasets.
- iii. The classification of plant diseases through the observation of visual symptoms on leaves is the focus of this research. This is important because leaf symptoms are often the first signs of any plant health concern. The investigation also described how other factors, such as environmental conditions and imaging quality, could affect the detection and classification of symptoms.
- iv. The study highlighted the compounding challenges of leaf disease classification considering that the symptoms of several leaf diseases can be similar, symptoms can vary between species, and distinguishing between healthy and diseased plants in real-life conditions can be challenging. These challenges were instrumental in defining the research agenda for enhancing classification models.
- v. It helps improve plant disease detection methodologies by proposing robust classifiers that can be used for plant disease detection under various environmental conditions and on various plant varieties.

1.4. Organization of Paper

The remainder of the essay is organised as follows: a review of the prior research literature is offered in the second part, along with an analysis of the research gaps and potential for more study. Here, every research article is summarised and a discussion is provided under the relevant category. The third section describes the deep learning framework and algorithm for the identification of healthiness and disease in plants. The fourth section presents the results of the evaluation and analysis. The fifth section summarises the contributions and identifies a few concerns for further research that have been raised by the studies before concluding.

2. Background and Related Work

Plant disease detection. It includes as:

- i. **Image-based approaches:** To extract characteristics from photographs of sick plants, these systems use several image-processing techniques. Subsequently, a classifier that can identify illnesses is trained using these characteristics.
- ii. **Symptom-based approaches:** These methods use the symptoms of a disease to detect and classify them. These include visual symptoms, such as leaf discoloration, as well as physiological symptoms, such as changes in plant metabolism.
- iii. **Genomic-based approaches:** These methods use genetic information, such as DNA or RNA sequencing, to detect and classify diseases. This can include identifying specific genes or markers associated with the disease [3].
- iv. **Machine learning-based approaches:** To categorise the illnesses that affect plants based on training data, these approaches make use of a variety of machine learning algorithms, such as SVM, Random Forest, Naive Bayes, and

CNN. Overall, many different approaches have been used for plant disease detection, but most research has focused on small datasets and specific species [4].

2.1 Previous Work

This section discussed the previous work that has been done Plant disease detection. The overall summary of the existing methods is listed in Table 1. The details of these methods are discussed in this section.

Research conducted by [5] outlines a method for identifying and recognising plant diseases. primarily focuses on fungus-related diseases, such as black spot and anthracnose. Convert the image's colour values into the colour space specified in the colour space transformation structure using L^*a^*b device-independent colour space transformation. Segmenting is performed on a picture after it has been transformed from the RGB to L^*a^*b colour space. Hierarchical Clustering is used for image segmentation. Feature vectors are generated using the colour moment method because they have the lowest feature dimensions and lowest computational complexity. For plant disease classification, the transformation of the feature vector is extracted from the leaf contour. By maximising the margin between the classes, the SVM identifies the optimal separating hyperplane.

The authors of [6] proposed a technique in which the input picture is transformed using the Hue Saturation Intensity (HSI) colour model. Hue, saturation, and intensity are the three characteristics that constitute the HSI colour model. A characteristic of a colour called a hue specifies the colour that the observer sees. Saturation is the measurement of white light combined with a hue. The intensity simply refers to the brightness or amplitude of the light. It drops off the intensity component, as it does not provide extra information regarding the analysis. After the colour space transformation, all the green-coloured pixels are masked based on the specified threshold value. All masked pixels were then completely removed for better disease classification. The affected leaf area was then removed and divided into equal-sized patches. Colour, texture, and feature analyses are carried out using spatial grey level dependence measures (SGDM). The grey-level co-occurrence method is a statistical method of describing the shape based on the relationship among the sampled gray levels in the image. A co-occurrence matrix was used to determine the number of different properties, including contrast, shadow, local homogeneity, and prominence. Two classification approaches were used for disease classification. The Minimum Distance Criterion (MDC) method was initially used to categorise diseases. Using the classification gain, which is 86.77%, the minimal distance classifier's performance is evaluated. a second classifier Support Vector Machine (SVM) that increases accuracy. The accuracy achieved in the case of the support vector machine was 94.74%. The SVM two-class classification technique is extended to multiclass classification to categorise into various classes.

This study, performed by [7], suggests a python-based technique for detecting plant diseases. It records a photograph of a sick leaf and provides the diagnosis. A convolutional neural network was used to categorise the illness. It is effective against common plant diseases, including rust, mildew, black spots, and bacteria. Before submitting the normalised photo to the CNN to determine the class designation, it first downscales the document image and normalises the pixel values.

This study, performed by [8], used a five-step process for disease classification. The first step is image acquisition, in which an image of various plant leaves suffering from a disease is taken. The next step includes image preprocessing, where low-light and low-resolution enhancement is performed by Convolutional Neural Network Layers. The third step is image segmentation which includes colour space transformation and segmentation. The YCbCr colour space model is applied to the picture once the RGB components have been transformed. YCbCr is the name given to the human vision-based colour representation model, which is independent of any particular device. The Y component, also known as the luma component, indicates luminance, whereas the Cb component represents chroma blue, and the Cr component represents chroma red. The next step is segmentation into k clusters, where k is the number of different components that need to be split. This is done using a genetic algorithm, which helps explore huge spaces as opposed to a brute force approach, which conducts a complete search and improves at every stage. The mathematical technique for extracting texture features employs GLCM, or Grey Level Co-occurrence Matrix, a statistical model for investigating object texture. It calculates various feature

vectors with features contrast, energy, dissimilarity, entropy, and correlation, and the computed values of the co-occurrence matrix are compared to the corresponding values in the training dataset feature values. A Support Vector Machine (SVM) is used to categorise the illness into its many subtypes. This SVM obtains an average accuracy of 83.16% across all of its kernels, including the Radial Basis Functional Kernel (RBF), linear kernel, polygonal kernel, and quadratic kernel.

This study [9] covers research that utilises image processing to identify plant diseases. Device-agnostic colour-space transformation was used in the first phase. To remove noise from the images, image preprocessing techniques were used, such as image clipping which crops the leaf image to find the region of interest, followed by image smoothing, which uses smoothing filters and image enhancement for enhancing the image's contrast. In the third step, picture segmentation, three algorithms were discussed: The Otsu Threshold method, together with Boundary and Spot Detection, K-means clustering, and both. The border of the diseased area is extracted using the boundary and spot detection approach, which also accounts for the pixels' eight connections. K represents the clustering strategy, which divides things into groups depending on how close they are to one another. Otsu's threshold segmentation technique creates a binary image from grey-level images by applying a specific threshold to convert all pixels whose value is less than the specific threshold to 0 and pixels whose value is greater than the particular threshold to 1. Otsu's method simply masks green-coloured pixels. Features such as colour, morphology, and texture edges are considered for plant disease detection.

The authors of [10] suggested a method for detecting cotton leaf disease. Bacterial blight, *Myrothecium*, and *Alternaria* were the three diseases discussed in this study. A low-pass filter, such as a Gaussian filter, was used to eliminate noise from the image. The Gaussian filter retains the edges corresponding to a high frequency. The active contour model approach allows the extraction of the area of interest from a picture. The snake-like active contour model fits the contours of the object and is an energy-minimising curve that is affected by outside limitations and image forces. It calculates invariant moments to extract the features. Seven invariant moments were obtained for training from three different kinds of pictures of the infected cotton leaf. Owing to its efficiency in multiclass classification issues where the target class is greater than two, the backpropagation neural network technique is used to classify plan illness. Levenberg–Marquardt optimisation is used for updating the bias and weight values in the discussed backpropagation neural network. The input and hidden layers of backpropagation each include seven nodes and utilise the tan-sigmoid function. The accuracy achieved by the backpropagation classification model used in this study was 85.52%.

This study, performed by [11], first discussed existing deep learning models and their applications. It also discusses many CNN architectures and compares their model sizes and input resolutions. Several distinct CNN layers were examined, including convolution, activation, pooling, and fully connected layers. The CNN layer discussed four hyperparameters for training: the number of filters, spatial extent, stride, and padding. The activation layer uses various nonlinear functions, such as tanh, sigmoid, and rectified linear unit functions, to introduce nonlinearity. With no change in depth, the pooling layer diminishes the spatial dimension. The size of the feature map is reduced using a pooling layer. A collection of high-level characteristics that are closely linked to the class or object was discovered by the fully connected layer. This layer classifies the image without considering the spatial relationship between the pixels known as the spatial relationship. Finally, the classifier comes in the role, and the classifier can perform binary or multiclass classification. For binary classification, the classifier uses the sigmoid activation function, and for multiclass classification, the classifier uses the softmax activation function. The study also used the CNN classification model to perform binary classification on tomato and apple leaves to classify whether a leaf is healthy or diseased. In this study, the CNN model used 64 input image sizes, a filter width of 3, and 0 numbers as padding with stride 1. The final model has a pooling layer and a ReLU activation function after each convolutional layer. To estimate the class of a particular image, the Softmax function was utilised with two fully connected layers. The model achieved an accuracy of 88.7.

This research was carried out by [12], who proposed a convolutional neural network-based deep learning model that can accurately assess an image's characteristics, recognise objects, and categorise them with little to no preparation. The model uses a convolution layer, pooling layer, activation layer, and feedforward layer, with their respective functions. This

study uses the linear vector quantisation (LVQ) method for training the data classification. In this study, owing to the linear vector quantisation method, all the CNN layers are not fully connected. Linear vector quantisation is a type of neural network that incorporates both supervised and competitive learning. This classification algorithm uses a heuristic method. The model is used in a variety of applications owing to its basic topology and adaptability. The input, Kohonen, and output layers are the three layers that make up the LVQ model. In the LVQ, the Kohonen and input layers are completely linked, but the Kohonen and output layers are not. The categorised results were provided to the output layer after learning in the Kohonen layer. In this study, 400 training and 100 testing images of tomatoes were used from the plant village dataset. Five different classes are used, with four classes of tomato diseases and one class for healthy tomato leaves.

Using supervised machine learning techniques such as Naive Bayes, Decision Trees, KNN, Support Vector Machines, and Random Forest, this research [13] provides a method for identifying maize plant diseases. This paper presents four classes of maize plant disease and works on a total of 3823 images of the maize plant. The RGB image is converted to grayscale in the preprocessing stage, and this study uses the label edge detection segmentation approach to extract the area of interest from the image. Segmentation may be performed in two ways: based on similarities and discontinuities. Image partitioning in similarity is based on predefined criteria, whereas image partitioning in discontinuities is based on abrupt changes in intensity. The gradient of the image intensities at each pixel inside the image was also calculated using the label edge detection segmentation approach. This study used RGB feature extraction to extract colour information and identify patterns. Naive Bayes achieved an accuracy of 77.46%; Decision Tree, 74.35%; K-nearest neighbour, 76.16%; and Random Forest, 79.23%; these were the classification methods that achieved the greatest accuracy.

The author of [14] proposed a system that uses image segmentation and machine learning to identify potato leaf diseases. It converts RGB images into HSV colour spaces and generates masks based on the HSV colour space information regarding colour, colour intensity, and colour brightness. It isolates the area of interest from the images by thresholding the HSV image for a range of green and brown hues. The green colour in the image signifies a healthy portion, and brown or any other colour signifies a diseased portion. Hu moments, Harlick Texture, and Color Histogram are three feature descriptors that are used in the article to extract features from the picture. Hu moments provide a feature vector for the shape. The texture feature was extracted from the picture using the Haralick texture feature descriptor. It was applied to grayscale images. The Harlick texture feature uses the gray level co-occurrence matrix (GLCM) for computing texture feature values. The colour intensity of an image is computed using the colour histogram descriptor. To categorise potato leaf disease, it employs seven classifiers: Random Forest, Logistic Regression, KNN, Decision Tree, Naive Bayes, Linear Discriminant Analysis, and SVM. It acquired an accuracy of 97% from Random Forest, 94% from Logistic Regression, 91% from K – Nearest Neighbour, 91% from Decision Tree, 84% from Naïve Bayes, 78% from Linear Discriminant Analysis, and 37% from Support Vector Machine. Among all the classifiers, the highest accuracy rate was achieved by random forest.

This study [15] proposed a method for the automatic detection of tomato leaf diseases. It worked on eight different categories of tomato leaf disease. For preprocessing of all the images in the dataset, Gaussian blur was used to minimise noise from the images. Haralick's Texture feature algorithm is used to extract texture information from the images. The mean, angular second moment, entropy, and many other attributes are considered to determine the contours and hierarchy of moments. Use of the ORB feature extraction algorithm. Machine learning techniques such as SVM, Logistic Regression, and Random Forest are used to categorise tomato leaf diseases. SVM produces better outcomes.

This study [16] presented a deep learning approach. A YOLOv3 object detector was used to isolate a leaf from the input image. The extracted leaf was analysed using several ResNet18 models. The model was trained using transfer learning. Sayad-Ab-Rahman et al. [17] For the classification of citrus illnesses and the diagnosis of plant diseases using images of leaves, a deep learning technique that is linked to the deep CNN model and is composed of two stages has been proposed. In the first stage, a region proposal network is used to identify the primary target area, and in the second stage, a classifier is used to assign the target area to the appropriate class. All images were adjusted to fit within the same intensity range using the histogram equalisation method.

This study [18] developed a method for classifying citrus diseases using feature fusion and transfer learning. Transfer learning was used to retrain the MobileNetv2 and DenseNet201 pre-trained models to provide feature vectors. The meta-heuristic whale optimisation technique was utilised to optimise the fused feature set. By using hybrid contrast stretching, the picture quality was enhanced. The accuracy of these findings was 95.7%. Deep learning algorithms such as Inception-v3, VGG-16, and VGG-19[19] were compared to machine learning techniques such as SVM, Random forest, and SGD for the identification of citrus plant diseases.

Colour, texture, and shape were used by [20] to identify diseases in plants. They discovered that the morphological outcome outperformed the other features. For feature extraction, two methods were discussed: the colour co-occurrence method and the use of H and B components to extract the colour of the leaves. The colour co-occurrence technique extracts a unique pattern for an image by taking colour and texture into account. The SGDM matrix was constructed to compute texture statistics, and the GLCM function was utilised to compute features. Leaf colour extraction may be used to identify a sick leaf's colour using the H and B components. The colour of a sick leaf is determined using an SOFM with a back propagation neural network. For the classification of diseases, backpropagation algorithms and Artificial Neural Networks (ANN) were investigated. The output of the ANN classification method is the weighted sum of the inputs after the neural network has been trained on the pictures in the training dataset. As a recurrent network, it uses a backpropagation neural network, and once the model has been trained, the weights are fixed and can be used for new queries.

The feature extraction approach uses HOG feature descriptors such [21] texture features and colour histograms. Papaya leaf images totalling 160 were used as the training set for the random forest classifier, which achieved an accuracy rate of 70.14%. Using an enhanced YOLOv4 model, a method for object detection [22] in a real-time context was suggested. To maximise the attained accuracy, changes were made to the present model. Under complicated orchard circumstances, verification techniques have been utilised for disease identification. It has been determined that pomegranate bacterial blight exists [23]. The picture was subjected to a grab-cut segmentation technique to distinguish between foreground and background. The backdrop and foreground of the image are separated by a rectangle. When you mask all of the background pixels, you separate them from the background. To determine which section is the background and which portion is the foreground, we consider the area outside the rectangle to be the background and the area within the rectangle to be the foreground. A colour distribution estimate is carried out using the Gaussian Mixture model, which assigns unknown pixels to the background or foreground depending on the degree to which they resemble the pixels that are immediately around them. Pomegranate edges were discovered using the Canny edge detection technique. To distinguish between pomegranates that are heavily contaminated and those that are not, the study also calculated the percentage of infection.

This study [24] uses an image enhancement approach to improve the captured pictures. To reduce noise, a median filter was used for all images before they were converted into the L^*A^*B colour space and given colour negation. To extract feature information from the leaves, a grey-level co-occurrence matrix was used. Deep neural networks (DNNs) have been used to classify illnesses. This study [25] used R-CNN and multitask learning to detect and categorise grape leaf diseases. Multilevel features, evaluation of special attention, and channel-wise attention are used to generate high-level feature maps.

Table 1: Comparative study of exiting work

Article	Architecture	Number of Images	Specie	Accuracy (%)
[25]	Custom	1796	Maize	96.7
[26]	CaffeNet	4483	Multiple	96.3
[27]	VGG16Net	13,262	Tomato	97.29
[28]	Custom	61,486	Multiple	96.46
[29]	Custom	800	Cucumber	94.9
[30]	Custom	3500	Wheat	81.04

[31]	GoogLeNet	3060	Maize	98.9
[32]	AlexNet	1450	Apple	97.3

Table 1 summarises the studies of several researchers, comparing their architecture with the number of images used for training and species with their accuracy.

3. Material & Methods

Figure 5 shows the process for the identification, classification, and forecasting of plant diseases, which includes the following steps:

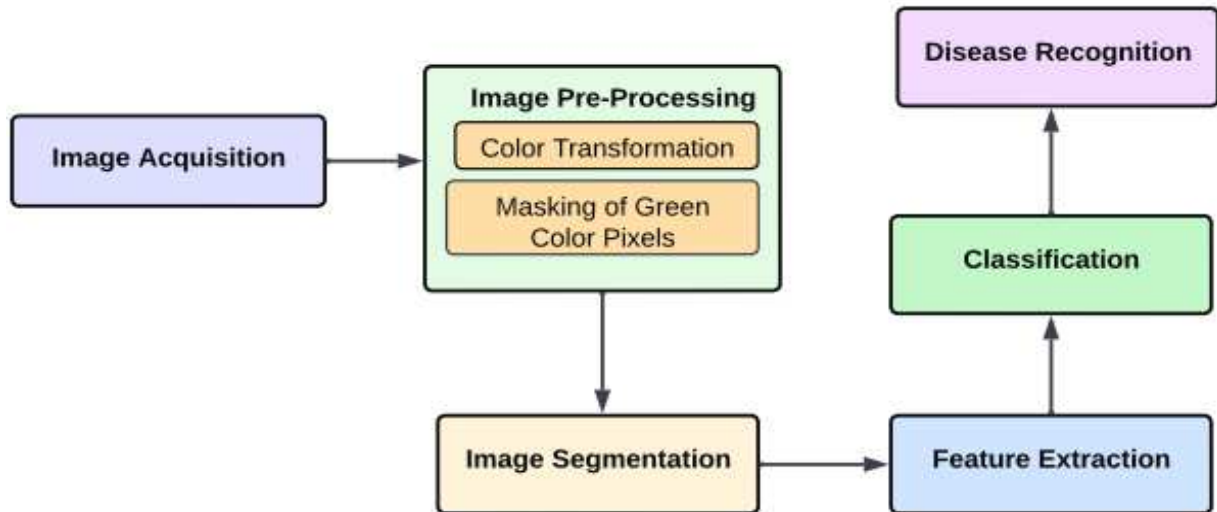


Figure 5: Proposed model architecture

3.1. Image Acquisition

The initial stage in detecting plant diseases is picture collection, which includes employing a digital camera to capture a range of plant images, turning those images into a dataset, or creating an online-accessible dataset with data on plant diseases. One folder that functions as a class label for categorisation should include photographs of the same kind of sickness. I have taken into account seven plant species, totalling 30 class designations.

3.2. Dataset

We considered seven plant species that will all have diseased and healthy classes. All species make up a total of 30 classes. In this dataset, all class labels make a sum of 53200 images. A tabular representation of the dataset is shown in Table 2.

Table 2: Dataset description

Species	Diseased	Healthy	Total
Apple	7204	2510	9714
Chilly	275	90	365
Corn	6821	2324	9145
Cotton	1277	426	1703
Grape	6912	2115	9027
Potato	4848	2280	7128
Total	13711	2407	16118

Total= 53200

3.3. Image Preprocessing

In this stage, before utilising any of the images for training or validation, they are all initially scaled to the same fixed image size of 256×256 . The procedure consists of two steps: all of the images in the first section have their colour spaces transformed using the device-independent HSV colour space. The HSV colour space transformation model consists of three components: hue, saturation, and value. These three components are as follows: hue is a measurement of how well an observer can distinguish colour. It is calculated as the colour angle on the vertical axis. Saturation is the proportion of the colour's purity or the quantity of white light combined with the colour. Its value ranges from 0 to 1. Pure colour has a value of 1, whereas completely impure colour has a value of 0. Value is a measure of the brightness or intensity of a colour. Its value ranges from 0 to 1. Value = 0 denotes the colour black, whereas value = 1 denotes the colour white. Masking the green pixels is the second step in the preprocessing module. Before utilizing any images for training or validation, all images are scaled to a fixed size of 256×256 pixels. The preprocessing procedure involved the following two steps:

Step 1: HSV Color Space Transformation

Initially, the images are transformed into a device-independent Hue, Saturation, Value (HSV) colour space. This transformation helps isolate colour information effectively. The HSV model consists of the following three components.

1. Hue (H) :

Hue represents the type of colour perceived by an observer. It is mathematically defined as a color angle along the vertical axis in the range $0^\circ \leq H \leq 360^\circ$.

$$H = \begin{cases} 0^\circ, & \text{if } \max(R, G, B) = \min(R, G, B) \\ 60^\circ \times \left(\frac{G - B}{\max(R, G, B) - \min(R, G, B)} \right) + \text{Offset}, & \text{otherwise} \end{cases}$$

Where R , G , and B are the red, green, and blue components of the image, and the offset depends on the sector of the hue calculation.

2. Saturation (S) :

Saturation measures the purity of the colour or the amount of white light mixed with it. Its value is calculated as

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)}$$

where $S \in [0,1]$. A value of $S = 1$ represents a pure color, while $S = 0$ indicates a completely impure (white) color.

3. Value (V) :

The value corresponds to the brightness or intensity of the colour. It is defined as:

$$V = \max(R, G, B)$$

where $V \in [0,1]$. A value of $V = 0$ represents black, and $V = 1$ represents white.

Step 2: Masking Green Pixels

To improve the disease detection accuracy, the green pixels of the leaf (assumed to represent healthy regions) are masked. This step involves identifying pixels with dominant green values and eliminating them to focus on areas of potential disease. The green mask is applied using the HSV space, where the range for the green hue is defined as

$$H_{\text{green}} \in [60^\circ, 180^\circ]$$

The reason for this is that while it is assumed that the green portion of the leaf is healthy, the green pixels do not provide any additional information about the disease. To obtain better results, we must mask the green pixels to eliminate the green portion. The figure displays an example picture in the BGR, RGB, and HSV formats, as shown in Figure 6.



BGR Image

RGB Image

HSV Image

Figure 6: The BGR, RGB, and HSV representations of images

3.4. Image Segmentation

The process of distinguishing objects from the background of an image is referred to as image segmentation. Before masking all the green pixels, which do not provide any more information about the sickness, we utilised a threshold-based approach to segregate the green and brown regions of the image. Figure 7 shows the segmented portion of the picture, with the leaf-free areas becoming dark.

**Figure 7:**

RGB Image

Segmented Image

Segmentation of leaf region

3.5. Feature Extraction

The basic technique for locating feature-related information, such as shape, size, colour, and texture information, is feature extraction. Feature-related information is very useful for computing the characteristics of an object or image. We can extract the image properties by stacking Hu moments, Haralick texture features, and colour histograms into a global feature vector.

1. Hu Moments for Shape Features

Hu moments are a set of seven invariant moments derived from image moments. These moments capture the shape characteristics of an object, which are invariant to scale, rotation, and translation. The general formula for image moments of order $(p + q)$ is given by:

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y)$$

where:

- $I(x, y)$ is the pixel intensity at position (x, y) ,
- p, q are the orders of moments.

The central moments μ_{pq} , which are invariant to translation, are defined as:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

where:

- $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the centroid coordinates.

From the central moments, Hu's seven invariant moments ϕ_i are calculated as non-linear combinations of the normalized central moments η_{pq} :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(1+(p+q)/2)}}$$

Hu's seven moments are then computed as:

Hu's seven moments are then computed as:

$$\phi_1 = \eta_{20} + \eta_{02}, \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \text{ (and so on for } \phi_3, \dots, \phi_7)$$

2 Haralick Texture Features

Texture features describe the spatial arrangement of pixel intensities. Haralick features are derived from the Gray-Level Co-occurrence Matrix (GLCM), G , which represents the frequency of pixel intensity pairs (i, j) separated by a specific distance and angle. The GLCM is defined as:

$$G(i, j) = \sum_{(x,y)} \begin{cases} 1 & \text{if } I(x, y) = i \text{ and } I(x + d_x, y + d_y) = j \\ 0 & \text{otherwise} \end{cases}$$

Common Haralick features are calculated from G as follows:

- Energy (Angular Second Moment):

$$\text{Energy} = \sum_i \sum_j G(i, j)^2$$

- Contrast:

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 G(i, j)$$

- Correlation:

$$\text{Correlation} = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) G(i, j)}{\sigma_i \sigma_j}$$

where μ_i, μ_j and σ_i, σ_j are the means and standard deviations of G along i - and j -axes.

- Homogeneity:

$$\text{Homogeneity} = \sum_i \sum_j \frac{G(i, j)}{1 + |i - j|}$$

3 Color Histograms

Colour histograms describe the distribution of pixel intensities across colour channels. For an image, the histogram $H(k)$ of color intensity k is defined as:

$$H(k) = \sum_x \sum_y \delta(I(x, y) - k)$$

where:

- $\delta(I(x, y) - k)$ is the Kronecker delta, which is 1 when $I(x, y) = k$, and 0 otherwise.

The histograms for Red, Green, and Blue (H_R, H_G, H_B) channels can be combined into a single feature vector:

$$\mathbf{H} = [H_R(k), H_G(k), H_B(k)] \text{ for } k = 1, \dots, n$$

Global Feature Vector

The global feature vector $\mathbf{F}$ is constructed by concatenating the features from all three methods:

$$\mathbf{F} = [\phi_1, \phi_2, \dots, \phi_7, \text{Haralick Features}, H_R, H_G, H_B]$$

This comprehensive representation ensures robust and diverse feature extraction for subsequent image analysis tasks.

3.6. Global Feature Vectors

Global feature vectors are a collection of data used to victories the entire picture. The form, colour, and texture of plant leaf features are the three aspects that we mainly focus on in this research. To create a single global feature vector for each picture, these three features were combined and superimposed. Haralick features are used to analyze textures, Hu moments are utilized as a shape descriptor, and color histograms are used to offer information on colors. These three feature descriptors all state the following:

1. **Hu Moments:** From central moments, which have predetermined rotational axes and dimensions, Hu moments are formed. The size, translation, and orientation of six of the seven Hu moments are invariant, but the seventh is skew-invariant, which means that the sign does not change for image reflection. Calculated Hu moments may be used to characterize the form of an image's objects. Hu moments are helpful if we want to compare the shapes of two objects to identify comparable objects in a picture.
2. **Haralick Texture Feature:** The Haralick texture feature is a method for encoding texture data in a picture. The grey level co-occurrence matrix (GLCM) is used to determine the characteristics of the hard lick texture. The connection between adjacent pixels is represented by GLCM. This resembles an adjacency matrix. The GLCM matrix counts the occurrences of pairs of neighbouring pixels with similar values. For texture analysis, four GLCM matrices were created. All four GLCM matrices' haralick texture features are computed to create four feature vectors, each of which has 14 dimensions, although 13 dimensions are considered for computational stability.
3. **Colour Histogram:** The colour histogram is a colour descriptor that shows an image's colour characteristics. It symbolises an even distribution of colours across a picture. It is a visual depiction of how many pixels fall within a certain colour spectrum.

3.7. Image Classification

Categorisation of the image into specific labels requires a classification procedure. We must train the classifier on our dataset to conduct the classification. Five classifiers were used to classify images into various categories, allowing us to assess how each model performs on the same dataset and improve our findings. The following is an explanation of the five classifiers.

3.7.1 Support Vector Machine

It is called a support vector machine in technical terms. It is a classification- and regression-supervised machine-learning algorithm. It can handle both linear and nonlinear separable data points. It uses a kernel function for nonlinear data points to convert a low-dimensional feature space to a high-dimensional feature space. To determine the class of an item, a border between the two classes is drawn. Each side of the hyperplane has a line drawn parallel to it; this line-to-line distance is known as the margin. Support vectors are the locations on each side closest to the hyperplane. For categorising

cases, the maximum margin hyperplane was used.

3.7.2 Logistic Regression

The statistical approach for categorising categorical class variables is called logistic regression. Our aim is to predict and classify plant diseases into targeted classes. Because there are more than two classes in this situation, multiclass logistic regression is used instead of a standard logistic regression technique, which can only handle binary classification. To calculate the probability of the target class, for each class I , the multiclass logistic regression trains n binary classifiers for n target classes. The following is the logistic regression hypothesis.

$$0 \leq h_{\theta}(x) \leq 1 \quad (1)$$

$$Y = \beta_0 + \beta_1 X \quad (2)$$

$$h_{\theta}(x) = \text{sigmoid}(Y) \quad (3)$$

$$\text{sigmoid}(Y) = \frac{1}{1 + e^{-Y}} \quad (4)$$

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \quad (5)$$

3.7.3 K-Nearest Neighbor

K-nearest neighbour is a supervised machine-learning technique that groups data points according to how similar they are. This presumes that linked things are located relatively close to one another. To calculate the similarity, the distance between the new data point and the data points in the training set was employed. The most comparable points are those that are at least K-minimum distances apart. Using the k training set data points with the greatest degree of similarity, it separates the new data point into k categories with the highest similarity. Classification tasks, such as determining the category to which a new data point belongs based on its similarities to prior data points in a labelled training set, are typical uses of KNN. The threshold at which a classification judgment must be made is set by the value of parameter K , which specifies the maximum number of neighbours to consider. Although lower values of K may be more accurate, they may also be more prone to overfitting the training data. Larger values of K will create smoother decision boundaries but may be less accurate for complicated classification tasks. Similar to other non-parametric methods, KNN does not presume anything about the underlying distribution of the data.

3.7.4 Random Forest

Random forest is a machine-learning strategy that develops a powerful classifier out of several poor ones, improving the model's performance. The supervised machine learning technique Rough Forest uses both the training and validation labels of the dataset. With the help of the subsets of the supplied dataset, many decision trees were produced. The class of the instance was anticipated using a majority vote of the prediction classes that each decision tree had predicted. Random forest is its ability to process high-dimensional data and identify relevant characteristics, even if they are only weakly linked with the objective variable. In addition, it is more robust against overfitting than the standalone decision trees. The technique used to construct a random forest model chooses the characteristics at each node of the decision tree at random. This is because each tree is trained with its unique data, which decreases their mutual dependence and boosts the model's precision.

3.3 Proposed Work

In this work, machine learning methods such as Wasserstein GAN and CNN are employed. To create fresh data samples that are analogous to a given set of training data samples, researchers have turned to the Wasserstein GAN, a form of generative adversarial network. To do this, it trains two networks, a generator network and a discriminator network, so that the generator network creates examples that are difficult for the discriminator network to distinguish from the original

training samples. Convolutional neural networks (CNNs), on the other hand, are specific types of neural networks that are frequently employed for image categorisation. Several convolutional layers are used to extract features from the input image before being followed by fully connected layers that perform classification using the features. To augment the original training data, the proposed method employs a Wasserstein GAN to produce synthetic data samples, with a CNN serving as the classifier. The data samples created were added to the original training data to create a larger dataset for CNN classifier training. Samples of test data were used to determine how well this method works.

3.3.1 Convolutional Layer

There are several convolutional layers; thus, we must specify how many filters each layer needs. To execute convolutional processes, a filter simply goes across each $N \times N$ block in the picture using an $N \times N$ matrix with randomly generated numeric values. We utilised a 7×7 filter size in our model. Here is an illustration of how a convolution layer with a 3×3 filter size and stride 2 works.

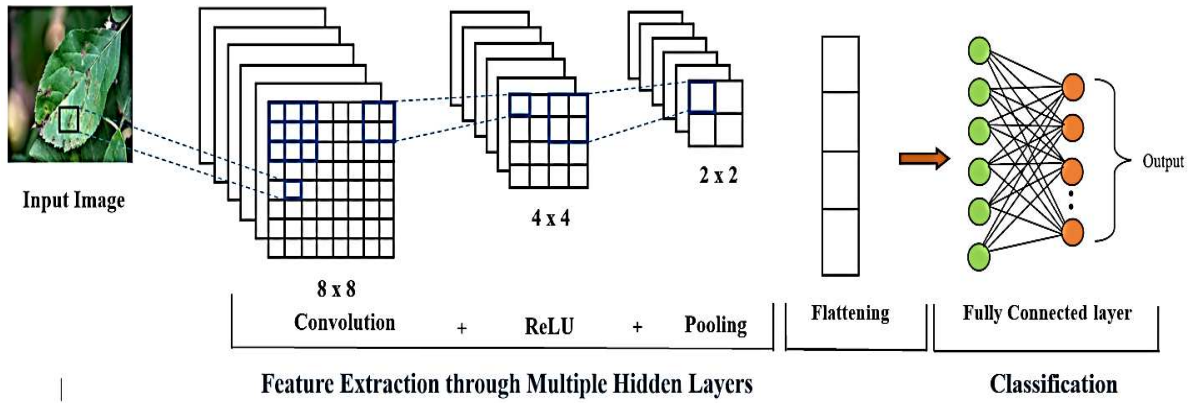


Figure 8: Proposed CNN architecture for crop classification

Using image data and convolutional neural networks (also known as ConvNet), these artificial neural networks, which have numerous hidden layers, are frequently trained as shown in figure 8. The Convolution layer, ReLU layer, and pooling layer are the three layers that make up the hidden layer. The hidden layer count was calculated using an image as the input. The feedforward neural network accepts the hidden layer's output as an input in the form of a single feature vector after the hidden layer flattens its output as part of the feature extraction phase. The feedforward network generates the result. Figure 9 depicts the action of the CNN model.

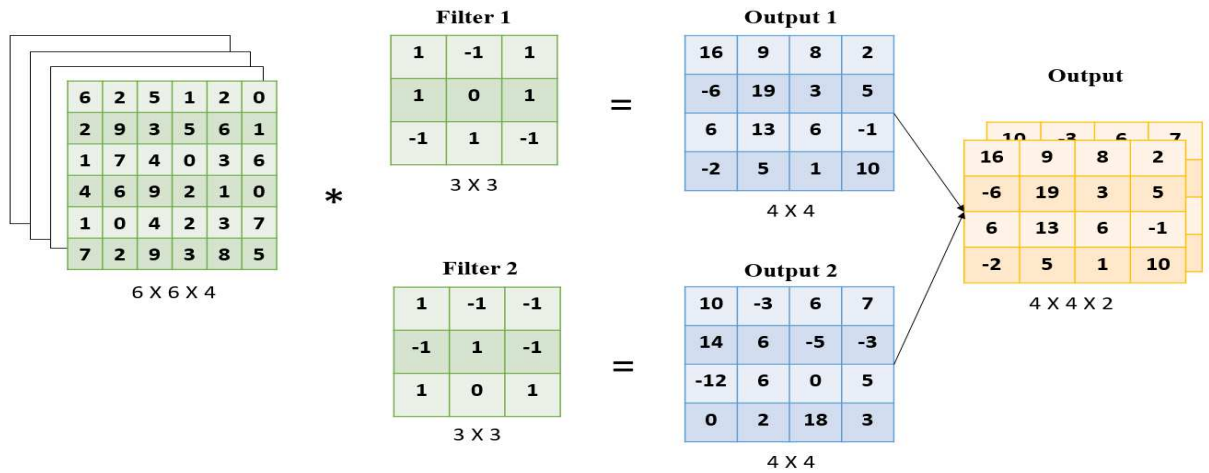


Figure 9: Convolutional layer operation

3.3.1 ReLU Layer

The convolution process produces feature maps, which are subsequently loaded into the ReLU layer. The rectified feature map of the input picture was obtained by activating the convolution layer with the ReLU activation function. The ReLU layer changed every negative value to 0. The ReLU layer renders the network less linear. It performs the element-by-element action illustrated in Fig. 10.

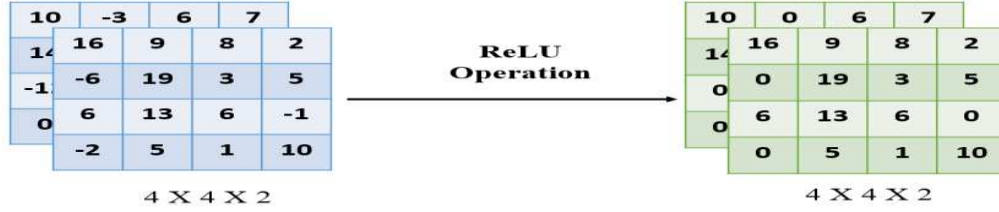


Figure 10: ReLU layer operation in proposed work

3.3.2 Pooling Layer

The dimension of the image is reduced by the pooling layer, which follows the convolution and ReLU processes. Simply stated, this involves downsampling the image and performing a pooling procedure. The rectified feature map of the ReLU layer is scaled down. The "max pooling" method uses the total of all values in Figure 11 to replace each N x N cell. A technique that flattens two-dimensional feature maps into a single continuous feature vector of integers is used for these data. Our simulation used a pool with a stride pattern of 2×2 .

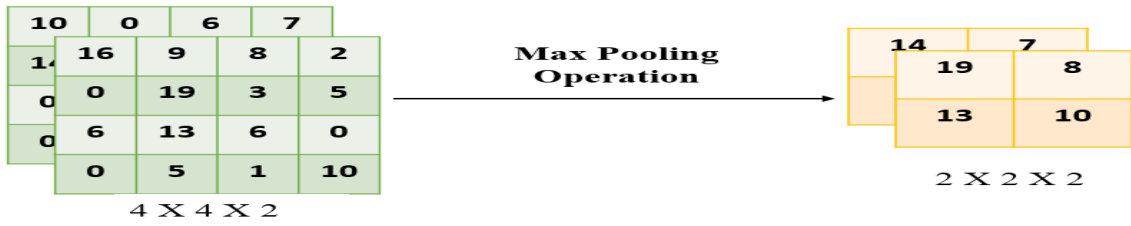


Figure 11: Pooling layer operation

This paper discusses the application of deep convolutional network (CNN) models for the identification of leaf symptoms. This is because these models have been successful in machine vision applications. A typical CNN model requires a greater number of parameters, which results in a greater cost associated with running the model. By using depth-separable convolution in place of conventional convolution in this investigation, we were able to reduce the number of parameters as well as the overall processing cost. Within the context of this application, models were trained using a readily available dataset that included information on 38 disease categories and 14 plant species.

3.3.3 Fully Connected Layer

A flattened feature vector is supplied to the fully connected layer's artificial neural network, which then adjusts various weights throughout specified iterations. Epochs are used to describe the iterations. The results of categorisation are output by this layer. The class of a particular leaf was predicted using the softmax function. To accomplish the classification, the flattened vector was subjected to the softmax function. We utilise the Softmax function because we have many output labels.

One definition of the Softmax Function is

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad \text{for } i = 1 \dots k \quad (6)$$

$$\text{Where } z = (z_1, \dots, z_k) \in R^k \quad (7)$$

We directly utilise the CNN classifier to extract features from the enlarged dataset that serves as its input. An end-to-end learning training procedure was used to describe this approach. The WGAN produces a dataset that is identical in size and form to the dataset it was fed as input. The original and generated data were combined to create the fusion dataset, which was then fed into the CNN classifier. The merged dataset is reduced in size owing to feature extraction from the convolution kernel and feature dimension from the pooling layer. An activation function was then used to classify the completely linked layer. Finally, the results for each class grade were used to establish the ultimate categorisation scenario.

$$y_i = \frac{e^{a_i}}{\sum_{k=1}^C e^{a_k}}. \quad (8)$$

3.3.4 Wasserstein Distance Functions

In this equation, the Wasserstein distance is given by (8). The generator's and discriminator's respective loss functions are given by Eqs.(9) and (10).

$$L(G) = -E_{x \sim P_\theta}[D(x)], \quad (9)$$

$$L(D) = -E_{x \sim P_r}[D(x)] + E_{x \sim P_\theta}[D(x)], \quad (10)$$

Where $D(x)$ denotes the output of the discriminator and P_r and P_θ denotes the distribution of the original data and the distribution of the created data, respectively.

During the WGAN optimisation process, the interaction between the weight constraint and cost function may cause the gradient to vanish or grow. The fundamental culprit is the WGAN gradient truncation process. Gradient truncation leads to a decrease in the model's capacity and the discriminating network's propensity to become a binary network [37]. The loss function of the discriminator when a gradient penalty is used instead of gradient clipping is expressed as follows:

where $\left[\|\nabla_x D(x)\|_p - 1\right]^2 \lambda$ is the gradient penalty weight parameter and where is the gradient penalty.

Because this issue falls under the category of multi-classification, the cross-entropy loss is used as the CNN classifier's loss function. The true label for sample point (x, y) , y . The value in a multiclass problem can only be a label set. Assume there are K label values and the probability that the i^{th} the sample is predicted as the k^{th} label value is $P_{i,k}$, i.e. $P_{i,k} = P_r(t_{i,k} = 1)$. has a total of N samples, and the model's loss function is written as follows:

$$L_{\log}(Y, P) = -\log \Pr(Y | P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k} \quad (11)$$

3.3.5 WG-CNN Optimizer

The WGAN model is optimised through the RMSProp approach, also known as the Root Mean Square Prop. The RMSProp approach employs the differential squared weighted average for the gradient of the weight W and offset b . The issue of the significant swing amplitude of the loss function during the update method must be addressed further. and the pace of convergence of the function must be accelerated. In the t^{th} iteration, the formula is (12)

$$\begin{aligned} s_{d\omega} &= \beta s_{d\omega} + (1 - \beta) dW^2, \\ s_{db} &= \beta s_{db} + (1 - \beta) db^2, \\ W &= W - \alpha \frac{dW}{\sqrt{s_{d\omega} + \epsilon}}, \\ b &= b - \alpha \frac{db}{\sqrt{s_{db} + \epsilon}}, \end{aligned} \quad (12)$$

Where $s_{d\omega}$ and s_{db} are the gradient momentums accumulated by the loss function in the preceding $t-1$ round iteration, β is an indicator of gradient accumulation, and is the network's learning rate. Unlike other optimisations, the RMSProp method computes the gradient differential squared weighted average. Using this method, which is helpful for swing amplitude correction, the swing amplitude of each dimension is decreased. However, the network function converges more rapidly.

The CNN model is optimised using the stochastic gradient descent technique (SGD), which adjusts the gradient and updates the weight from the previous iteration together with the weights for the network model. The entire process can be

shown by (13).

$$\begin{aligned} V_{t+1} &= \mu V_t + \alpha \nabla L(W_t), \\ W_{t+1} &= W_t - V_{t+1}, \end{aligned} \quad (13)$$

W_{t+1} is the network weight after the $t + 1^{th}$ iteration, and V_{t+1} is the network weight's update amount in the $t + 1^{th}$ iteration.

All the layers in the network typically have a gradient and non-linear activation units associated with them, and in the convolution layers, there may be a normalising layer and a pooling layer. It begins with a series of five convolution layers, then moves on to three fully linked layers, and finishes with a softmax layer. Each of the first two convolution layers (conv1, 2) is followed by normalisation and a pooling layer, whereas the last convolution layer (conv5) is followed by a single pooling layer. Our modified version of the softmax layer receives input from the fc8 layer, which has 38 outputs (one for each class in our dataset). Finally, the input from (fc8) is exponentially normalised by the softmax layer, yielding a distribution of values over all 38 classes. These numbers reflect the degree to which the network agrees that a certain class accurately describes an input image. While the first two fully connected levels (fc6, 7) each have a dropout layer with a dropout ratio of 0.5, the first seven layers each have an ReLu nonlinearity activation unit.

We used a convolutional neural network (CNN) with cross-entropy (CE) loss in this investigation. This loss penalises the predicted probability corresponding to a ground truth class while ignoring the interclass connection. The CNN models AlexNet, ResNet50, and EfficientNet-b1 were initialised using their own architectures and pre-trained weights from ImageNet. An effective strategy for using pre-trained models is to first train a model on a fake task using extensive picture datasets. This allows the model to acquire general image characteristics that can then be readily applied and refined for new characterisation tasks. Transfer learning is used to leverage the information already acquired by pre-trained models for rice categorisation. A portion of this code immobilises the first layers to preserve the acquired characteristics after regenerating them using the rice dataset for a few further levels. This approach enables the efficient execution of training even with a limited amount of data. Model optimisation was achieved by adjusting the hyperparameters. The factors considered were the learning rate, batch size, optimiser selection, regularisation methodology, and number of training epochs. Random search approaches were used to tweak the hyperparameter grid optimisation.

The models were trained using the prepared dataset and optimised hyperparameters. The training process involved feeding batches of training images into the models, computing the cross-entropy loss, and updating the model weights through backpropagation and gradient descent optimisation. The dataset we used was preprocessed with rice grains, as highlighted in the image, and the background was black. The total number of data points was 75000, with 15000 data points belonging to each class. The dataset was divided into three parts.

- i. Training: 62000 with 12400 belonging to each class.
- ii. Validation: 5000, with 1000 belonging to each class.
- iii. Testing: 8000, with 1600 belonging to each class.

For performance evaluation of the three CNN models, they were used to train the data for running 20 epochs using a learning rate of $1e-5$ and their respective number of parameters. Because the parameters for the employed models are different from each other, in this context, the first model, AlexNet, is employed which contains 62.3 million parameters and is also known as a simple convolutional neural-based backbone model. Feature extraction of the task was performed using this model after each layer. The obtained features were then flattened for further classification. This is one way in which the training and testing accuracies can be obtained., ResNet50, which has 23 million parameters, was used for the same node. This model has a residual-based backbone that considers the residues and ensures that the layers of the model do not hinder the performance of each other. In other words, it does not have any features that reduce the performance of the previous layer, even if it has a very deep network. Finally, the EfficientNet-b1 variant, which has 7.8 million parameters, was used. This model considers vertical, horizontal, and other features, as well as the final prediction.

After training all models with the same hyperparameters and number of epochs on the same setting of training data, a comparative study of all three CNN variants was performed. Despite the many differences in the number of parameters, the premise of this study was to establish that architecture, feature extraction, and generalisation play a more important role than the number of parameters. Even with 7.8 million parameters, EfficientNet-b1 performs well owing to its better feature extraction technique. From these results, it can be inferred that, even with a smaller number of parameters, it is possible to match the performed models with better feature extraction techniques and better and lighter architectures. From the results, it can be concluded that EfficientNet-b1 can be considered the model in which the best trade-off between complexity and performance is found. Even with a small number of parameters, it outperformed the AlexNet model and matched the performance of ResNet50.

Another point to be highlighted in the paper is that EfficientNet-b1 is considered the best model because the entire dataset images used are of rice grains with different qualities; hence, it is quite difficult to classify the types of rice through manual processes. With the pre-processed images, the need for a proper CNN model has been identified by considering the proper dimensions and angles of every rice image dataset to properly classify them, as the images are very close to each other. The model called InceptionNet paved the way and also performed well in terms of extraction of the horizontal and vertical features. Convolutionally, a final feature extraction process and a final classification process were performed. A model with a smaller number of parameters performs well in the feature extraction processes.

Convolutional Layer as

$$(I * F)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot F(m, n) \quad (1)$$

where i, j are the spatial dimensions of the output feature map, and m, n are the dimensions of the filter. Rectified Linear Unit (ReLU) is defined as

$$\text{ReLU}(x) = \max(0, x) \quad (2)$$

Then, Max pooling, a common pooling operation, is defined as

$$\text{MaxPooling}(I)(i, j) = \max_{a \in W_{i,j}} a \quad (3)$$

where $W_{i,j}$ is the window of inputs covered by the pooling operation at position (i, j) .

The hybrid model combines the features from AlexNet, ResNet50, and EfficientNet-b1. Let F_{AlexNet} , F_{ResNet50} , and $F_{\text{EfficientNet-b1}}$ represent the feature extraction functions of the respective models. The combined feature vector F_{combined} can be represented as:

$$F_{\text{combined}} = \text{Concatenate}(F_{\text{AlexNet}}(I), F_{\text{ResNet50}}(I), F_{\text{EfficientNet-b1}}(I)) \quad (4)$$

where I is the input image, and Concatenate is the operation that merges the feature vectors from each model. After feature combination, fully connected (FC) layers were used for classification. The output of the FC layer is

$$\text{FC}(x) = Wx + b \quad (5)$$

where x is the input vector to the FC layer, W is the weight matrix, and b is the bias vector. The final output layer, typically a softmax layer for multiclass classification, computes the probability distribution over classes as follows:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}} \quad (6)$$

where z_i is the input to the softmax function for class i , and k is the total number of classes. The model is trained by optimising a loss function, commonly the cross-entropy loss for classification, which is defined as

$$\text{Cross-Entropy}(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \quad (7)$$

where y is the true label in one-hot encoding, and $\hat{y}$ is the predicted probability distribution from the softmax layer.

When the complexity of the three models is considered, the number of model parameters strictly decreases. The number of parameters in ResNet50 was three times lower than that in AlexNet, and in the case of EfficientNet-b1, it was approximately two and a half times less than that of ResNet50 and five times smaller than that of AlexNet. The performance of the models was comparatively closer, but the complexity was higher for the other two models. Although there is a vast

difference between the number of parameters, the performance obtained is considered to be the best in the case of the EfficientNet-b1 model owing to its low complexity. In addition, this model can be run easily on a CPU rather than using HPC, such as TPU's and GPUs, to classify them. Therefore, this is the performance of the models used.

The impact of hybridization

The present study proposes a CNN model that is a hybrid of AlexNet, ResNet50, and EfficientNet-b1, with the intention of merging the strengths of each network into a single network. The fundamental idea behind this is to consolidate the characteristics of several designs to enhance the accuracy of the classification and resilience of the structure. The structure, data flow, and optimisation strategies used in the construction of the hybrid model are discussed in the next section. The pictures taken by the input layer were 224×224 pixels, and they had three colour channels that represented RGB. It is essential to prepare the data to improve the generalisability of the model. The normalisation of pixel values on a common scale and additional enhanced data, which may be obtained by rotation, flipping, zooming, or cropping, are included. Overfitting can be avoided by augmenting the data, which is particularly important when considering the small variance between the types in the dataset.

Each input image X with dimensions $H \times W \times C$ (Height, Width, Channels) is normalized to pixel values in the range $[0,1]$. Let the normalized input image be

$$X' = \frac{X}{255}$$

We also apply data augmentation $A(X)$ to increase generalization:

$$X_{\text{aug}} = A(X')$$

This augmented image X_{aug} is the input to all three CNN models.

AlexNet, ResNet50, and EfficientNet-b1 are three parallel CNN architectures that feed pictures created from the beginning. The inputs are processed individually by each model, which then extracts distinctive characteristics relevant to categorisation.

The AlexNet algorithm was chosen for this project because it is a straightforward and effective method for extracting fundamental characteristics, such as edges and textures. The spatial dimensions were downsampled using max-pooling, which was interspersed across the five convolutional layers constituting this structure. This form of architecture was used to apply filters to identify low-level patterns within the photographs, such as colour gradients and edges, which are crucial characteristics for distinguishing between many types of rice.

The convolutional operation in AlexNet can be expressed as

$$F_{\text{Alex}} = \sigma(W_{\text{Alex}} * X_{\text{aug}} + b_{\text{Alex}})$$

where $*$ denotes convolution, W_{Alex} is the filter weight matrix, b_{Alex} is the bias, and σ is the ReLU activation function:

$$\sigma(z) = \max(0, z)$$

It is now time for feature maps to undergo the process of flattening their vectors and going through completely linked layers with ReLU activation. To avoid overfitting, a dropout layer is used, which involves random deactivation of neurones throughout the training process. It is expected that the output of AlexNet will be a feature vector that represents fundamental picture features. The issue of vanishing gradients, which is often associated with deep neural networks, was addressed by ResNet50 using residual connections. Because of these connections, the network can skip layers, which makes the gradient flow during the backpropagation process easier to manage. This makes it possible to train extremely deep designs. Twenty-five layers comprised the ResNet50 architecture, with each layer comprising a significant number of convolutional layers, which were then followed by batch normalisation and ReLU activation in the residual blocks. The use of residual connections by ResNet results in an improved gradient flow. A residual block is responsible for

$$F_{\text{Res}} = \sigma(W_1 * X_{\text{aug}} + b_1) + X_{\text{aug}}$$

During the backpropagation process, the skip connection guarantees that the gradient flow is adequately maintained, which prevents vanishing gradients from occurring. However, because it can capture high-level and sophisticated traits such as form and detailed patterns, it is a great option for the categorisation of rice varieties that are visually similar to one another. In addition to performing global average pooling, ResNet50 constructs a feature vector that comprises deep learned patterns from the pictures. Compound scaling is used in EfficientNet-b1, which aims to maintain equilibrium between depth, width, and resolution to achieve the highest possible performance, while simultaneously reducing the amount of processing resources required. This design uses depth-wise separable convolutions to significantly reduce the number of parameters while maintaining the same level of performance. These Squeeze-and-Excitation (SE) blocks were responsible for the recalibration of channel-wise feature responses. This allows the network to concentrate more on the portions of the picture that are the most significant. In EfficientNet-b1, SE blocks and depth-wise separable convolutions are used in the learning process. Iterative Convolution with Depth.

$$F_{\text{Eff}} = \sigma(W_d * X_{\text{aug}} + b_d)$$

where W_d is the depthwise filter. The SE block scales the feature channels as follows:

$$F_{\text{Eff,SE}} = F_{\text{Eff}} \cdot \sigma(W_{\text{SE}} F_{\text{Eff}} + b_{\text{SE}})$$

Consequently, the model's emphasis on essential characteristics was further highlighted.

Therefore, it is particularly suitable for extracting fine-grained characteristics from complicated datasets, such as distinguishing between rice types with only slight visual changes. To build a compact feature vector, the feature map of EfficientNet-b1 is transferred to global average pooling, which then constructs the feature vector. Concatenation is then performed on the vectors obtained by AlexNet, ResNet50, and EfficientNet-b1 to create a single universal vector. Therefore, the significance of a feature concatenation layer lies at the core of merging all three designs for general strengths. These architectures include AlexNet, which provides fundamental texture information; ResNet50, which provides complicated patterns and forms; and EfficientNet-b1, which provides fine-grained optimised features.

In order to create a unified feature vector, the output feature vectors F_{concat} from the three different models are concatenated together

$$F_{\text{concat}} = [F_{\text{Alex}}, F_{\text{Res}}, F_{\text{Eff,SE}}]$$

This vector combines both low- and high-level characteristics to obtain a more accurate categorisation.

The concatenated vector encodes both high- and low-level characteristics; as a result, it enhances the capability of the model to accurately identify the five different types of rice included in the dataset. Because of this integration, all the designs are able to mix together and play the strengths of the three, which ultimately results in a performance that is superior to that shown by the separate models. Subsequently, the concatenated feature vector is transmitted across layers that are completely linked. The ReLU activations included in each of these are responsible for modifying the feature combinations to extract meaningful representations. The use of a dropout layer with a dropout rate of 0.5 helps the model generalise to data that it has not seen before and avoids overfitting. Subsequently, it passes the concatenated feature vector through a layer that is completely linked.

$$z = W_{\text{fc}} F_{\text{concat}} + b_{\text{fc}}$$

where W_{fc} and b_{fc} are the weights and biases of the fully connected layer. The final output was computed using the softmax function:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^n \exp(z_j)}$$

where $\hat{y}_i$ is the predicted probability for class i and n is the total number of classes.

The softmax method was used in the output layer to provide the probability for each of the five rice classifications. In multiclass classification, the softmax function guarantees that the probabilities for each rice-type class amount to one. To tune the hybrid CNN model, the Nadam optimiser, an adaptive learning rate algorithm that integrates RMSprop and momentum-based gradient descent, was used. A learning rate of 1e-5 was used for this purpose in conjunction with

categorical cross-entropy loss. This was implemented because it represents the most suitable solution for multiclass classification. Subsequently, the model underwent training for a total of 20 epochs with a batch size of thirty-two, aiming to attain a balance between convergence speed and model stability. Moreover, L2 weight regularisation can mitigate overfitting by imposing penalties on excessively large weights. Additionally, to enhance the generalisation capability of the model, data augmentation techniques such as CutMix and Mixup are used. This procedure was performed to achieve the intended outcomes. The performance of the hybrid model was assessed using many metrics, including accuracy, precision, recall, and F1-score. It was designed to comprehend the errors produced by the model and evaluate its strengths and limits. Confusion matrices were then generated. In this matrix, a more precise categorisation outcome relative to the individual models may be observed for minor variations in rice types.

A categorical cross-entropy loss function was used for the model training.

$$L = - \sum_{i=1}^n y_i \log (\hat{y}_i)$$

where y_i is the true label (one-hot encoded) and $\hat{y}_i$ is the predicted probability for class i .

The Nadam optimizer updates the model's parameters θ using the gradients of the loss function:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t + \epsilon}}$$

where η is the learning rate, m_t and v_t are the first and second moment estimates, and ϵ is a small constant to prevent division by zero.

To prevent overfitting, L2 regularisation was applied to the following weights:

$$L_{\text{total}} = L + \lambda \sum_k \|W_k\|_2^2$$

where λ is the regularization coefficient and W_k are the model weights.

Additionally, dropout is applied with a probability p to deactivate neurons during training:

$$F_{\text{drop}}(x) = \begin{cases} 0 & \text{with probability } p \\ \frac{x}{1-p} & \text{otherwise} \end{cases}$$

According to the results, the hybrid model outperformed all separate CNN architectures tested. It demonstrates much greater accuracy while generally providing a superior model with fewer false positives and negatives, resulting in an improvement in the overall performance of the job. The merging of complementary strengths from AlexNet, ResNet50, and EfficientNet-b1 allowed the maintenance of state-of-the-art performance in rice categorisation tasks.

1. The combination of the simplicity of AlexNet, residual learning from ResNet50, and the compound scaling approach from EfficientNet-b1 resulted in superior accuracy compared to any individual model used in isolation. This hybrid methodology allowed us to capture both the aerial landscape and individual trees more effectively.
2. Increased generalisation: Because machine learning contributes to half of the issue solutions, it exhibits superior generalisation to unfamiliar data. The implementation of a hybrid model enables the effective use of many architectures that can acquire knowledge from a wider range of characteristics, thereby enhancing the generalisation ability of diverse datasets.
3. The models exhibited enhanced efficiency in feature extraction by collectively addressing distinct tasks and assigning varying degrees of importance to each other. For instance, ResNet50 was designed to prioritise the screening of deeper features, whereas EfficientNet-b1 was specifically designed to handle well scales ranging from 150 to 600. This resulted in improved classification outcomes.

4. One tradeoff between hybrid models is their high computational cost. To address this, we used EfficientNet-b1 as a lightweight base architecture to reduce the computational footprint. This compromise between computational efficiency and model performance is feasible.

We contend that the performance of the CNN can be improved by using a more effective loss function. On the other hand, to ascertain the degree of separation that exists between two distributions, a well-known statistic called the Wasserstein distance (WD) is applied. Direct solutions to the WD issue require an excessive amount of time to implement, whereas iterative techniques can save money but have several problems, including computational instability and difficulties in parameter selection. In this study, we demonstrate that the WD problem has a closed-form solution, which we refer to as the "Hierarchical Wasserstein CNN," when two distributions are present in a data field that is arranged hierarchically. Using this framework, we can construct unique loss functions that solve the concerns present in the current widespread approaches to loss. The multi-CNN information fusion that we carried out specifically for this purpose forms the basis of our approach for the construction of category hierarchies. A binary tree is often used when there is a need to demonstrate the semantic links that exist between different classes. The next step is to train a hierarchical Wasserstein Convolutional Neural Network (HW-CNN), which is a variant of CNN that uses HWD as its loss function. This allows the acquisition of rich feature data. We have used HWD-CNN-based loss for a given image I belonging to the k^{th} class, the HWD-CNN loss is defined as

$$\mathcal{L}_{\text{HWD-CNN}}(\omega; I) = \sum_{C \in \mathcal{F}} \left[\sum_{i \in \mathcal{L}_C} (p_i - p'_i) \right]^2 e_C \quad (14)$$

where

$$p'_i = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise.} \end{cases}$$

Now the hybrid loss uses HWD-CNN computational values are normalized as

$$\mathcal{L}_{\text{hybrid}}(\omega; I) = -\log p_k + \lambda \sum_{C \in \mathcal{Y}} \left[\sum_{i \in \mathcal{L}_C} (p_i - p'_i) \right]^2 e_C \quad (15)$$

where $\lambda \in \mathbb{R}^+$ is the parameter value.

4. Result & Analysis

The previous section dealt with the research methodology used to identify plant diseases based on the symptoms that appeared on the leaf surface. The methodology explains the various steps for extracting useful features for disease identification and classification models.

4.1. Implementation Tool Used

The research methodology was implemented using Python. A 10-fold cross-validation technique was used to evaluate the accuracy of all machine learning models.

4.2. Result Analysis

The results of the implementation are described in this section. We conducted an experiment using 53200 image from our collection, which covers the most prevalent bacterial, viral, and fungal infections. First, a 256×256 fixed size was applied to each picture. SVM, Logistic Regression, Random Forest, KNN, and CNN, a deep learning technique, were the four machine learning methods we used for classification. For the Machine learning classification operation, we first applied the HSV colour transformation model to convert all images to the HSV format. We masked all green-value pixels using a particular threshold value and removed the background from the images. Figure 7 shows the HSV colour transformation of the sample image. We applied a global feature descriptor to extract features from the images for classification. Then, we performed classification using SVM, Logistic Regression, Random forest, and KNN.

We trained the proposed model on the same dataset's 53200 image, which included 30 classes. Our model has 11,206,302 parameters, of which 11,196,702 are trainable and 9600 are not. Our model had a 97.34% accuracy rate after 60 training epochs. The number of epochs increases the training and validation accuracy while decreasing the training and validation loss. The training and validation accuracy curves are shown in Figure 12. Figure 13 depicts a graph of the training and validation losses.

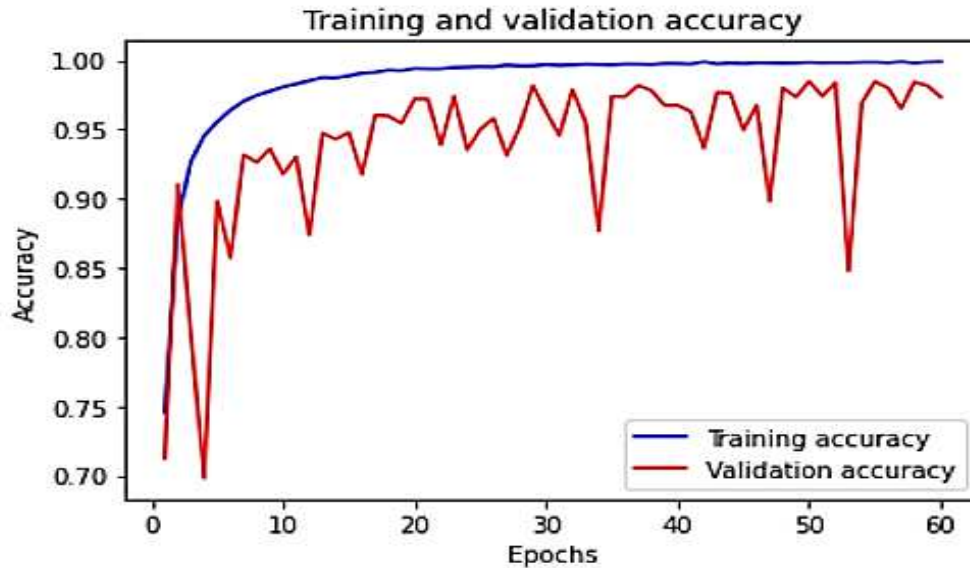


Figure 12: The suggested Work's training and validation accuracy graph

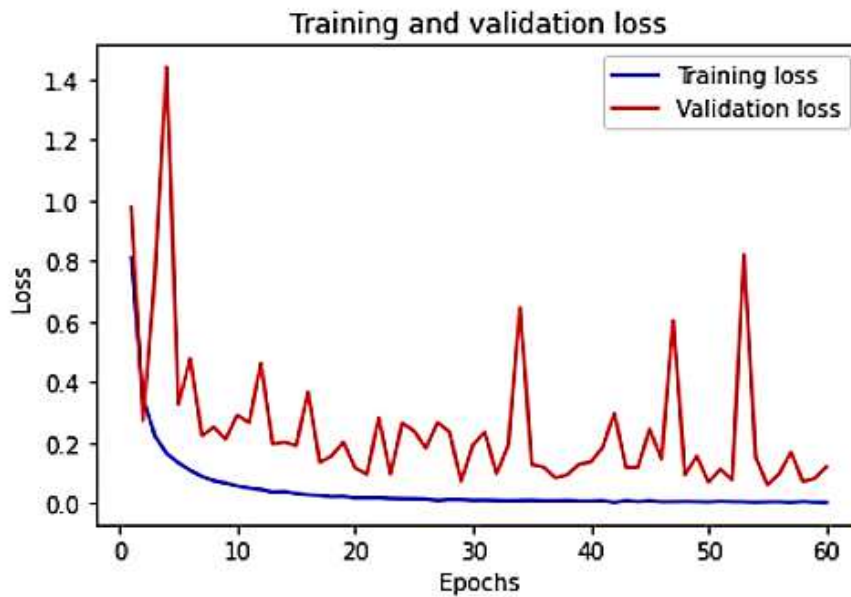


Figure 13: The suggested Work's training and validation loss graph

The classification accuracy of all four machine learning models was determined using a 10-fold cross-validation technique. Table 3 and Figure 14 show the accuracies of all five classification models.

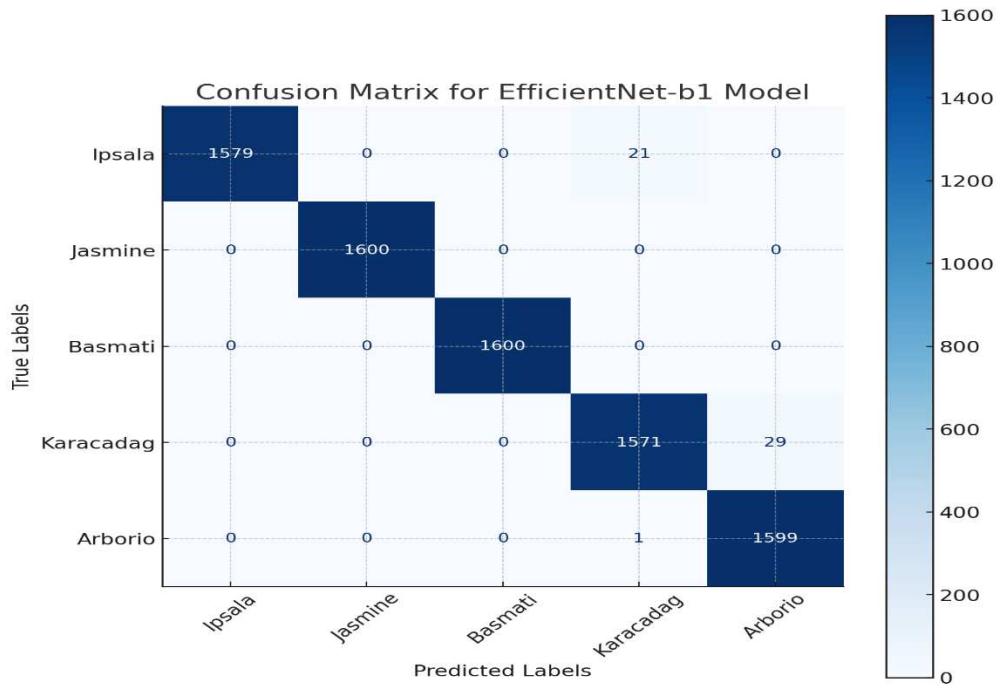


Figure 14: Confusion matrix for the EfficientNet-b1 model, visualizing the true versus predicted labels for each class

The confusion matrix for the EfficientNet-b1 model is shown in Figure 14, revealing the machine learning model's classification performance across the five classes of the dataset: Ipsala, Jasmine, Basmati, Karacadag, and Arborio. The entries represent the counts of true labels by predicted labels where true labels are the rows and predicted labels are the columns. High values for all classes are given as labelled in the format of diagonal entries which tell correct classifications reducing the Error. Non-diagonal entries indicates misclassification, that are insignificant. As an example: Ipsala class: 21, Karacadag class: 29 This visualization highlights the great performance of the model, as it correctly classifies all images, with very few errors.

Table 3: Accuracy achieved by all models

Models	Accuracy(%)
CNN[14]	97.34
RF[15]	96.91
KNN[16]	84.01
SVM[17]	74.75
LR[18]	72.18
Proposed Model	99.19

In the table 3 various ML models accuracy are compared for plant disease classification. While the Proposed Model shows superior performance in terms of accuracy achieving the highest accuracy of 99.19% among existing models. The CNN model is second at 97.34%, and Random Forest (RF) at 96.91%. The accuracy of K-Nearest Neighbor (KNN) shows a sizable decline in performance at 84.01% and even lower performanceS with Support Vector Machine (SVM) at 74.75%. The accuracy of Logistic regression (LR) is 72.18%, which gives lowest accuracy. The results underlined that the proposed model outperformed all other models, yielding approximately 1.85% greater performance than CNN and 26.01% performance than LR, which proved that the proposed model can be fully use.

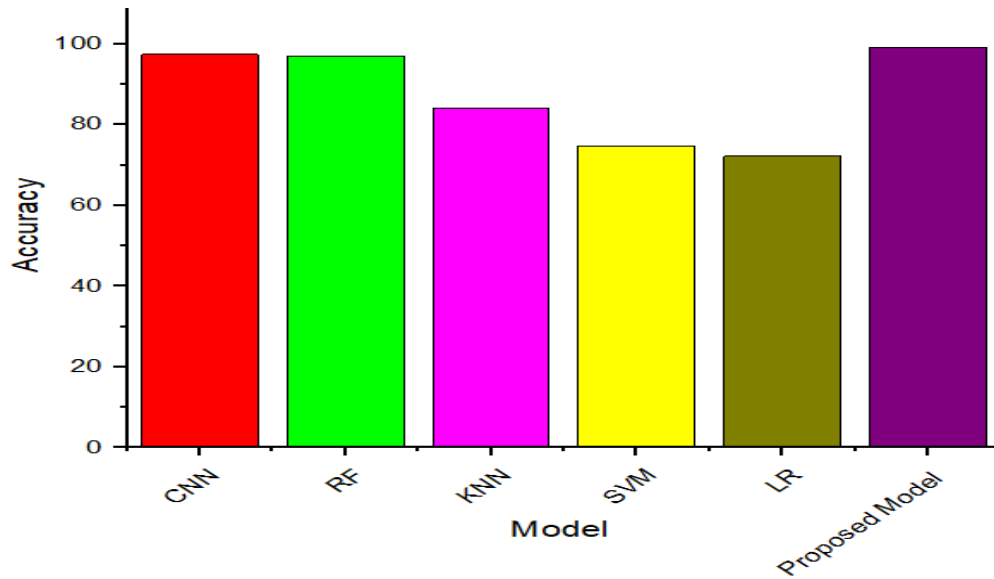


Figure 14: Comparative Analysis of Existing Model concerning their accuracy

We first done a prediction on 5 images of five categories of plant leaves, namely chilly leaf spot, corn common rust, cotton bacterial blight, cotton leaf curl virus, and a healthy leaf image of potatoes. The predicted result shows the output label of the leaf and the category of the leaf to which the leaf belongs, such as bacterial, fungal, or viral. These are the output of the predictions of five images. Table 4 shows the predicted results of the five types of leaf images.

Table 4: Experiment results for predication of Disease identified on the dataset

Image	Disease Identified	Predictions
	Chilly_leaf spot	Fungal
	Corn_Common_rust	Fungal
	Cotton_bacterial_blight	Bacterial
	Cotton_curl_virus	Viral
	Potato_healty	Healthy

We also performed predictions on 50 plant images using our CNN model. Table 5 shows the results of our predictions.

Table 5: Prediction results of 50 images of plant leaves

Disease Names	Number of Images	Correctly Classified	Incorrectly Classified	Category
Apple Cedar Rust	4	4	0	Fungal
Apple Scab	3	2	1	Fungal
Chilly Leaf Spot	5	3	2	Fungal
Corn Common Rust	3	3	0	Fungal
Chilly Whitefly	4	2	2	Viral
Cotton Healthy	4	4	0	Healthy
Cotton Leaf Blight	4	4	0	Bacterial
Potato Early Blight	5	5	0	Fungal
Potato Healthy	2	2	0	Healthy
Tomato Early Blight	6	4	2	Fungal
Tomato Healthy	4	4	0	Healthy
Tomato Yellow Leaf Curl	6	6	0	Viral

A comparison was made between the results of the proposed model and those of existing models. It has been shown that the proposed model (Wasserstein GAN and CNN) outperforms state-of-the-art models with a remarkable accuracy of 99.19%.

Table 6: Classification of Test images using the proposed method

			
cherry (Infected)	Apple (Healthy)	Grape(Healthy)	cherry(Healthy)
			
maize(Healthy)	Corn(Infected)	soybean(Healthy)	potato(Infected)

Table 6 shows the test results for the different plant leaf classifications.

Table 7: Comparative analysis of existing method with proposed work.

Author	Method	Accuracy
[18]	Fast Enhanced Learning Method +KNN	90%

[19]	K-mean clustering + GLCM + SVM	89.8%
[20]	Based on univariate statistical features test + SVM	99%
Proposed Model	Wasserstein GAN and CNN	99.19%

Table 7 summarises the comparisons between the proposed model (Wasserstein GAN and CNN) and the methods described in [18], [19], and [20]. In contrast to conventional approaches, the proposed method yields more accuracy. It is clear from Table 7 that there have been improvements made when the proposed model is utilised, further demonstrating the strength of the model as it compares against several popular methods used for classification purposes in related studies. The approach presented in [18] uses a fast enhanced learning method with a K-nearest neighbour (KNN) classifier, achieving an accuracy of 90%. KNN is a simple yet powerful algorithm used for classification based on data point similarity. Its performance is highly dependent on the quality of the features extracted, although it works well on smaller datasets; training on larger datasets is often computationally intensive. The method presented in [19] combines K-mean clustering for segmentation, grey-level co-occurrence matrix (GLCM) for texture feature extraction, and a Support Vector Machine (SVM) for classification with a performance of 89.8%. It relies on a combination of clustering and texture-based feature extraction, but suffers owing to the linear separability characteristic of SVM in complex datasets and the manual tuning required for optimal clustering.

Kummar et al. [20] selected a subset of relevant features using univariate statistical feature tests and applied SVM for classification. The formulated method resulted in an accuracy of up to 99%, showing that a reasonable combination of good feature selection and classifier can lead to a significant performance improvement. The proposed WGAN and CNN models outperformed these methods, attaining 99.19% accuracy. WGAN provides high-quality synthetic data that alleviates data imbalance problems and improves robustness. The CNN learns to extract hierarchical features from images and applies classification. Generative Modelling-Based Deep Learning: It provides better generalisation and accuracy than classical approaches by combining the strengths of generative modelling and DL. The findings show that the proposed model achieves a better performance than previous methods, particularly when dealing with complex datasets and obtaining high precision in classification tasks.

5. Conclusion

Using deep learning and specifically CNNs for plant disease identification is a promising approach that could potentially help farmers detect diseases early and take appropriate measures to prevent damage to their crops. Replacing standard convolution with depth-separable convolution is a good idea because it can reduce the number of parameters and computation cost, which can be beneficial for training and inference speed. Additionally, using an open dataset with a variety of plant species and disease classes can help ensure that the models are robust and generalise well to different scenarios. This adjustment decreases the dependence of the model on a large number of factors and its computational burden. To train their model, we used information from a dataset that included 38 disease categories and 14 plant species. To estimate the likelihood of leaf illness, they employed a convolutional neural network (CNN) with a Wasserstein Distance (WD) for cross-entropy (CE) loss. This method improves the model's performance across a wide range of parameters by allowing the model to efficiently construct an appropriate loss function. The model was trained using an HW-CNN, which is a hierarchical Wasserstein convolutional neural network. An HWD loss function was used in the HW-CNN, which is a version of the standard CNN. According to the findings, the new method outperforms prior methods with an accuracy of 99.19%. Deep learning techniques may be used to detect and categorise agricultural illnesses, which is especially important in poor nations, where it can have a major influence on crop productivity and food security if caught early. To assist farmers

in taking the required steps to minimise crop loss and boost productivity, the suggested method provides a more efficient and accurate way to diagnose leaf diseases.

Author Contribution

Daya Shankar Verma led the conceptualization and methodology development, particularly for the HW-CNN architecture and Wasserstein metrics. Jitendra K. Mishra provided crucial oversight, refining experimental design and validating results. Ankit Kumar was responsible for implementing and optimizing the HW-CNN models, including extensive coding, training, and data analysis. Linesh Raja contributed to the literature review, data collection, and initial data labeling, while Vimal Dungdung assisted with performance evaluation and result interpretation. Achintya Singhal supported manuscript writing and editing, ensuring academic standards and clear presentation of findings.

Funding

The authors have no specific funding for this study.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest.

References

- [1] M. Afrokh *et al.*, "Ethnobotanical, phytochemical and anti-oxidant study of fifty aromatic and medicinal plants," *Chemical Data Collections*, Data Paper vol. 43, 2023, Art no. 100984, doi: 10.1016/j.cdc.2022.100984.
- [2] S. Afzal *et al.*, "Grasp of wheat leaf rust through plant leaves extract and bioagent as an eco-friendly measure," *Journal of King Saud University - Science*, Article vol. 35, no. 2, 2023, Art no. 102452, doi: 10.1016/j.jksus.2022.102452.
- [3] M. Al-Fatimi, "Ethnopharmacological survey of endemic plants used in ethnomedicinal knowledge of Soqatra Island," *Journal of Ethnopharmacology*, Article vol. 304, 2023, Art no. 116033, doi: 10.1016/j.jep.2022.116033.
- [4] S. Bensaadi and A. Louchene, "Low-cost convolutional neural network for tomato plant diseases classification", *IAES International Journal of Artificial Intelligence*, Article vol. 12, no. 1, pp. 162-170, 2023, doi: 10.11591/ijai.v12.i1.pp162-170.
- [5] M. N. Bhargavi and P. J. Devang, "A Review on Some Indian Medicinal Plants useful in Hair Care", *Current Traditional Medicine*, Review vol. 9, no. 2, pp. 141-145, 2023, Art no. e100522204558, doi: 10.2174/2215083808666220510165110.
- [6] I. E. Cock, N. A. Ndlovu, and S. F. Van Vuuren, "The traditional use of southern African medicinal plants to treat oral pathogens, and studies into their relevant antimicrobial properties," *South African Journal of Botany*, Review vol. 153, pp. 258-279, 2023, doi: 10.1016/j.sajb.2023.01.003.
- [7] H. Cole, "The organ supply chain: Geography and the inequalities of transplant logistics," *Transactions of the Institute of British Geographers*, Article vol. 46, no. 4, pp. 1008-1021, 2021, doi: 10.1111/tran.12458.

- [8] P. V. Dang *et al.*, "Study on the chitinase-induced efficiency against anthracnose on soybean plant by oligochitosan-Zn²⁺ complexes," *Case Studies in Chemical and Environmental Engineering*, Article vol. 7, 2023, Art no. 100285, doi: 10.1016/j.cscee.2022.100285.
- [9] M. S. Deelip and K. Govinda, "ExpSFROA-Based DRN: Exponential Sunflower Rider Optimization Algorithm-Driven Deep Residual Network for the Intrusion Detection in IOT-Based Plant Disease Monitoring," *International Journal of Semantic Computing*, Article vol. 17, no. 1, pp. 5-31, 2023, doi: 10.1142/S1793351X22400165.
- [10] L. G. Gemin *et al.*, "Polysaccharides combined to copper and magnesium improve tomato growth, yield, anti-oxidant and plant defense enzymes," *Scientia Horticulturae*, Article vol. 310, 2023, Art no. 111758, doi: 10.1016/j.scienta.2022.111758.
- [11] P. S. Georgantopoulos, D. Papadimitriou, C. Constantinopoulos, T. Manios, I. N. Daliakopoulos, and D. Kosmopoulos, "A Multispectral Dataset for the Detection of Tuta Absoluta and Leveillula Taurica in Tomato Plants", *Smart Agricultural Technology*, Article vol. 4, 2023, Art no. 100146, doi: 10.1016/j.atech.2022.100146.
- [12] K. Halteh, K. Kumar, and A. Gepp, "Using Cutting-Edge Tree-Based Stochastic Models to Predict Credit Risk," *Risks*, Article vol. 6, no. 2, Jun 2018, Art no. 55, doi: 10.3390/risks6020055.
- [13] P. P. Hein, K., Arunachalam, Y. Fu, M. Zaw, Y. Yang, and X. Yang, "Diversity of medicinal plants and their therapeutic usages of Kachin people (Jinghpaw) in the central part of Kachin State, Myanmar," *Journal of ethnopharmacology*, Article vol. 302, no. Pt B, p. 115921, 2023, doi: 10.1016/j.jep.2022.115921.
- [14] M. T. Hulin *et al.*, "Genomic and functional analysis of phage-mediated horizontal gene transfer in *Pseudomonas syringae* on the plant surface," *New Phytologist*, Article vol. 237, no. 3, pp. 959-973, 2023, doi: 10.1111/nph.18573.
- [15] N. Kilic, H. Ayvaci, M. E. Güldür, and M. Dikilitas, "First report of 'Candidatus Phytoplasma australasia'-related strain' (16SrII-D) in *Pyracantha angustifolia* (narrowleaf firethorn)," *Australasian Plant Disease Notes*, Article vol. 18, no. 1, 2023, Art no. 1, doi: 10.1007/s13314-022-00486-8.
- [16] S. Kumar Sahu and M. Pandey, "An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model," *Expert Systems with Applications*, Article vol. 214, 2023, Art no. 118989, doi: 10.1016/j.eswa.2022.118989.
- [17] A. Pal and V. Kumar, "AgriDet: Plant Leaf Disease severity classification using agriculture detection framework," *Engineering Applications of Artificial Intelligence*, Article vol. 119, 2023, Art no. 105754, doi: 10.1016/j.engappai.2022.105754.
- [18] S. Phanthavong *et al.*, "First report of *calonectria pseudoreteauidii* in Lao PDR associated with a leaf spot disease of *macadamia integrifolia*", *Australasian Plant Pathology*, Note vol. 52, no. 1, pp. 23-26, 2023, doi: 10.1007/s13313-022-00900-w.
- [19] H. Raju and V. K. Narasimhaiah, "Insights on assessing image processing approaches towards health status of plant leaf using machine learning," *IAES International Journal of Artificial Intelligence*, Article vol. 12, no. 2, pp. 884-891, 2023, doi: 10.11591/ijai.v12.i2.pp884-891.
- [20] R. Ramamoorthy, E., Saravana Kumar, R. C. A. Naidu, and K. Shruthi, "Reliable and Accurate Plant Leaf Disease Detection with Treatment Suggestions Using Enhanced Deep Learning Techniques," *SN Computer Science*, Article vol. 4, no. 2, 2023, Art no. 158, doi: 10.1007/s42979-022-01589-w.
- [21] S. Ranjana *et al.*, "Hippophae salicifolia D. Don, a Fascinating Medicinal Plant: An Update on its Traditional Medicinal Uses, Ethnopharmacology and Phytochemistry," *Current Traditional Medicine*, Short Survey vol. 9, no. 2, pp. 19-32, 2023, Art no. e270522205357, doi: 10.2174/221508380866220527144311.
- [22] Y. K. Rathore, R. R. Janghel, S. K. Pandey, A. Kumar, K. U. Singh, and M. A. Shah, "A Novel AI-Based Approach for Better Segmentation of the Fungal and Bacterial Leaf Diseases of Rice Plant," *Journal of Sensors*, Article vol. 2022, 2022, Art no. 6871085, doi: 10.1155/2022/6871085.
- [23] S. Sabreen *et al.*, "Antifungal Potential of Selected Medicinal Plants Against *Candida albicans* and HPLC Analysis,"

-
- Current Bioactive Compounds*, Article vol. 19, no. 2, pp. 49-55, 2023, Art no. e200622206189, doi: 10.2174/1573407218666220620114814.
- [24] S. Santiteerakul, A., Sopadang, K. Y. Tipayawong, and K. Tamvimol, "The Role of Smart Technology in Sustainable Agriculture: A Case Study of Wangree Plant Factory," *Sustainability*, vol. 12, no. 11, Jun 2020, Art no. 4640, doi: 10.3390/su12114640.
- [25] J. Scanavacca *et al.*, "Antimicrobial activity of Tetradenia riparia leaf essential oil," *Boletín Latinoamericano y del Caribe de Plantas Medicinales y Aromáticas*, Article vol. 22, no. 2, pp. 255-267, 2023, doi: 10.37360/blacpma.23.22.2.19.
- [26] G. Sharma, A. Kumar, N. Gour, A. K. Saini, A. Upadhyay, and A. Kumar, "Cognitive framework and learning paradigms of plant leaf classification using artificial neural network and support vector machine," *Journal of Experimental and Theoretical Artificial Intelligence*, Article 2022, doi: 10.1080/0952813X.2022.2096698.
- [27] M. Tahir, H. Asnake, T. Beyene, P. Van Damme, and A. Mohammed, "Ethnobotanical study of medicinal plants in Asagirt District, Northeastern Ethiopia," *Tropical Medicine and Health*, Article vol. 51, no. 1, 2023, Art no. 1, doi: 10.1186/s41182-023-00493-0.
- [28] V. C. R. Viana *et al.*, "Green Nanobioinsecticide of a Brazilian endemic plant for the *Aedes aegypti* control," *Sustainable Chemistry and Pharmacy*, Article vol. 32, 2023, Art no. 100992, doi: 10.1016/j.scp.2023.100992.
- [29] M. S. Watt *et al.*, "Prediction of the severity of Dothistroma needle blight in radiata pine using plant based traits and narrow band indices derived from UAV hyperspectral imagery," *Agricultural and Forest Meteorology*, Article vol. 330, 2023, Art no. 109294, doi: 10.1016/j.agrformet.2022.109294.
- [30] P. V. Yeswanth, R. Khandelwal, and S. Deivalakshmi, "Super Resolution-Based Leaf Disease Detection in Potato Plant Using Broad Deep Residual Network (BDRN)", *SN Computer Science*, Article vol. 4, no. 2, 2023, Art no. 112, doi: 10.1007/s42979-022-01514-1.
- [31] S. Yu, L. Xie, and Q. Huang, "Inception convolutional vision transformers for plant disease identification," *Internet of Things (Netherlands)*, Article vol. 21, 2023, Art no. 100650, doi: 10.1016/j.iot.2022.100650.
- [32] L. Zhao *et al.*, "Pathogenicity and diversity of large-spored *Alternaria* associated with three solanaceous vegetables (*Solanum tuberosum*, *S. lycopersicum* and *S. melongena*) in China," *Plant Pathology*, Article vol. 72, no. 2, pp. 376-391, 2023, doi: 10.1111/ppa.13665.