

Leaf Image-based Plant Disease Identification using Color and Texture Features

Nisar Ahmed

University of Engineering and Technology

Hafiz Muhammad Shahzad Asif

University of Engineering and Technology

Gulshan Saleem

COMSATS University Islamabad - Lahore Campus

Muhammad Usman Younus (✉ usman1644@gmail.com)

Universite Federale Toulouse Midi-Pyrenees Ecole Doctorale Mathematiques Informatique
Telecommunications de Toulouse <https://orcid.org/0000-0001-9033-1767>

Sadia Anwar

Aarhus University: Aarhus Universitet

Muhammad Rizwan Anjum

Islamia University of Bahawalpur: The Islamia University of Bahawalpur Pakistan

Research Article

Keywords: feature extraction, classification, plant disease, leaf image and leaf disease, disease identification

Posted Date: April 20th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-438951/v1>

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

[Read Full License](#)

Leaf Image-based Plant Disease Identification using Color and Texture Features

Nisar Ahmad¹, Hafiz Muhammad Shahzad Asif², Gulshan Saleem³, Muhammad Usman Younus^{4*}, Sadia Anwar⁵, Muhammad Rizwan Anjum⁶

1 Department of Computer Engineering, University of Engineering and Technology Lahore, 54890, Pakistan.

2 Department of Computer Science, University of Engineering and Technology Lahore, 54890, Pakistan.

3 Department of Computer Science, COMSATS University Islamabad (Lahore Campus), 54890, Pakistan.

4 Ecole Mathématiques, Informatique, Télécommunications de Toulouse, Université de Toulouse, France.

5 Department of Business Development and Technology, Aarhus University, Herning, Denmark.

6 Department of Electronic Engineering, The Islamia University Bahawalpur, 63100, Pakistan.

*Corresponding Author: Muhammad Usman Younus. Email: usman1644@gmail.com

Abstract

Identification of plant disease is usually done through visual inspection or during laboratory examination which causes delays resulting in yield loss by the time identification is complete. On the other hand, complex deep learning models perform the task with reasonable performance but due to their large size and high computational requirements, they are not suited to mobile and handheld devices. Our proposed approach contributes automated identification of plant diseases which follows a sequence of steps involving pre-processing, segmentation of diseased leaf area, calculation of features based on the Gray-Level Co-occurrence Matrix (GLCM), feature selection and classification. In this study, six color features and twenty-two texture features have been calculated. Support vector machines is used to perform one-vs-one classification of plant disease. The proposed model of disease identification provides an accuracy of 98.79% with a standard deviation of 0.57 on 10-fold cross-validation. The accuracy on a self-collected dataset is 82.47% for disease identification and 91.40% for healthy and diseased classification. The reported performance measures are better or comparable to the existing approaches and highest among the feature-based methods, presenting it as the most suitable method to automated leaf-based plant disease identification. This prototype system can be extended by adding more disease categories or targeting specific crop or disease categories.

Keywords: feature extraction, classification, plant disease, leaf image and leaf disease, disease identification.

1. Introduction

Inconsistency and delay in the identification of plant diseases cause a reduction in the quantity and quality of yield. Losses due to plant diseases or other pest account for 20 to 40% of global annual productivity [1]. Studies have been carried out to assess the estimated loss caused by different diseases [1]. Yield loss also contributes toward increased consumer prices and a drop in the earnings for crop producers. Accurate and timely identification of plant diseases is crucial for ensuring maximum yield and is beneficial for farmlands in remote areas.

Advancements in machine vision have made it possible to perform the tasks of visual identification and these visual recognition methods can be employed for successful identification of plant diseases [2]. Image-based disease management and surveys have a long history of more than 90 years when

aerial images were used to study crop disease [3]. Disease detection and identification have improved since then and informative and sophisticated analysis is being carried out [4]. Image-based disease identification is under continuous development and recent reviews can demonstrate this [5-7]. Moreover, studies have shown that image processing methods are effective to identify plant species or diseases from leaf images [5-7]. Imaging devices have become cheaper and common with better quality images. Reliability, accuracy and precision of machine identification tasks have also continued to improve. As 60-70% of diseases appear exclusively on plant leaves which paves a way towards leaf image-based plant disease identification [8].

This research is aimed at providing a solution for small farm holders in remote areas where expert knowledge is unavailable. The proposed work presents an approach to design a system for disease identification. The proposed work can be transformed into a mobile application that can perform the task of disease identification through visual classification. The major reason for crop yield loss is the attack of pests or various diseases so the risk of crop loss can be reduced by providing a solution of disease identification in farms [9]. The main objectives of the work are to identify whether a leaf is diseased or healthy and classify the type of the disease to a diseased leaf image. The objective is fulfilled based on visual classification and can be accomplished in scenarios where the leaf disease is visually distinguishable. The previous research in this domain is intended to identify the diseases from a few crops or diseases with small sample size. Therefore, they provide a solution of limited utility as plants are present in great variety. The PlantVillage is a dataset with 54,309 leaf samples which provides 38 class labels for 14 crops [10]. This public dataset has a large training set with class labels which can be used by many researchers.

A feature-based approach is proposed here which tries to overcome the limitations of previous work in terms of accuracy and portability and the essence of the research lies in provision of high accuracy and portability to mobile devices. The proposed approach is a prototype system that is based on PlantVillage database having 38 classes. The system can identify the leaf as healthy and diseased and then categorize it into one of the diseases, present in the database. To identify additional symptoms or diseases, such cases need to be added to the database, and retraining the system will enable to identify those disease categories as well.

2. Related Work

Different approaches to identification and quantification of plant disease are in practice and leaf image-based identification of plant disease is one of them [11-17]. It is by far the easiest way to automatically identify plant diseases and can be used for the identification of various diseases [12]. The occurrence of plant disease causes specific changes in the texture and color of the leaf and therefore leaf imagery can be used to extract color and texture-based features to train a classifier. Some of the significant literature in the domain of plant leaf-based disease identification is provided below.

There are two approaches for leaf image-based plant disease identification: (i) deep learning-based, which uses complex architectures to automatically learn features (ii) feature-based, which extract hand-crafted features such as color and texture features to train a conventional machine learning algorithm. The deep learning-based approaches have provided higher accuracies but they require more computation and therefore not suitable for mobile or handheld devices with limited memory and computations. Some of the designed systems are targeted different diseases of some specific plant, whereas the other approaches target multiple plant diseases. Phadikar et al. [18] have presented a feature-based approach to disease identification of rice crop. They have used Fermi energy-based method for segmentation followed by color, contour and locality mapping. Rough set theory is used for selection of important features and rule mining with 10-fold cross-validation is used for system testing. Baquero et al. [19] have presented a Content-Based Image Retrieval (CBIR) system which uses color structure descriptors and nearest neighbors to classify important diseases or disease symptoms such as chlorosis, sooty molds and early blight. Similarly, Patil et al. [20] has also pre-

sented a CBIR and extracted color, shape and texture-based features. Sandika et al. [21] have proposed a feature-based approach for disease identification of grapes leaves. They have also performed the comparison of texture feature's performance. Oberti et al. [22] has targeted the fungal disease of grapevine plant (powdery mildew) due to its adverse effects on the crop yield and quality of produce. They have used multi-spectral imaging and captured grapevine leaf images at a range of angles (0 to 75 degrees). They have also highlighted that the detection sensitivity increases with the increase in angle and the highest value is obtained at 60 degrees and for early middle ages, the sensitivity improves from 9-75% with change in angle from 0-60 degrees. Similarly, Zhang et al. [15] have presented a feature-based approach that transforms the image into superpixel representation and then segments the desired region using k-means and extracts Pyramid of Histogram of Orientation Gradient (PHOG). Sharif et al. [23] has presented a feature-based approach for citrus fruit plant disease. They have used a hybrid feature selection technique based on principal component analysis and feature statistics. Singh et al. [24] has also presented a feature-based approach for pine trees. Bai et al. [25] has targeted cucumber plant disease and proposed an improved fuzzy c-mean based clustering technique to segment the diseased leaf area. Hlaing et al. has presented a feature-based approach and used PlantVillage dataset. They have extracted SIFT features and applied generalized pareto distributions to calculate density function. Support vector machines is used to train on these features and provided a 10-fold cross-validation accuracy of 84.7%.

Recently deep learning-based approaches are used for leaf image-based plant disease identification. Picon et al. [26] have used CNN for classification of plant diseases and claimed that deep learning-based methods have provided the highest performance. They have used Deep Residual Neural Networks and mainly targeted septoria, tan spot and rust. Ferentinos et al. [12] has also presented a deep learning-based solution using an extended PlantVillage dataset with 58 classes. They have used pre-trained VGG for transfer learning and provided an accuracy of 99.53%. The cross-dataset evaluation has provided a sharp decrease of 25-35% in accuracy indicating poor generalization. Mohanty et al. [14] and Yuan et al. [27] have proposed deep learning-based solutions using PlantVillage dataset. They have used pre-trained CNN and applied transfer learning to classify plants into 38 classes. Zeng et al. [28] have presented a high-order residual CNN architecture that extracts low-level details as well as high-level abstract representation simultaneously to improve classification performance and provided 91.3% classification accuracy with good generalization performance.

The proposed approach has targeted the problem through a more suitable two-step approach. PlantVillage dataset with 38 classes is used to demonstrate the proposed approach, and extended version of the dataset is also reported by some researchers but it was not available publicly at the time of this research. The first step is identification of the leaf as healthy or diseased. This step is performed with the bag-of-features approach which is an effective method of visual classification. Further processing will only be performed on the diseased leaf to identify the type of disease affecting the plant leaf. This identification is performed by segmenting the diseased leaf region and extracted color and texture features. It is to be noted that we have extracted a comprehensive set of texture features which is not used in the literature for disease identification task. Feature normalization and selection are performed to obtain the most discriminatory feature set for classification. Five classification algorithms are tried in the final stage and Support Vector Machines (SVM) with cubic kernel is used as a final classifier. The proposed approach has provided comparable results to the state-of-the-art algorithms and demonstrates the effectiveness of texture-based visual classification algorithms for the task of disease identification.

3. Materials and Methods

Hughes et al. [8] have released a comprehensive dataset of plant diseases, expertly curated for leaf image-based plant disease identification. The dataset is named as PlantVillage and contains 54,309

images of diseased and healthy plant leaves and contain 38 class labels. These images cover 14 crop species: tomato, strawberry, squash, soybean, raspberry, potato, bell pepper, peach, orange, grape, corn, cherry and apple. These leaf images are affected by fungal, bacterial and viral disease. The dataset also contains images of 12 healthy species: apple, bell-pepper, blueberry, cherry, corn, grape, peach, potato, raspberry, soybean, strawberry and tomato. The leaf images are of 256×256 pixels' dimension with RGB colors.



Fig. 1 Some randomly selected leaf images from PlantVillage Dataset

Fig. 1 shows some sample leaf images from the dataset, these images are drawn randomly from the dataset to look at the raw form of images. These images are captured by trained staff in a regularized process and therefore contrast adjustment, color-cast and background removal are necessary to remove any possible bias.

Table 1 highlights the 38 classes of PlantVillage dataset. There are 26 diseases occurring in 11 different plants and 12 healthy plant categories. The system would be able to identify any other disease or plant category if the suitable number of training images are provided before final model construction. It is important to highlight that the proposed system will only be able to categorize the image when the images are visually distinguishable as it follows a visual classification approach.

Table 1 Plant and their corresponding disease covered by the PlantVillage database

Category	Plant	Disease
1	Apple	Scab
2		Black Rot
3		Cedar Apple Rust
4	Bell Pepper	Bacterial Spot
5	Cherry	Powdery Mildew
6	Corn	Cercospora Leaf Spot / Gray Leaf Spot
7		Common Rust
8		Northern Leaf Blight
9	Grape	Black Rot
10		Esca (Black Measles)
11		Leaf Blight (Isariopsis Leaf Spot)
12	Orange	Haunglongbing (Citrus Greening)
13	Peach	Bacterial Spot
14	Potato	Early Blight
15		Late Blight

16	Squash	Powdery Mildew
17	Strawberry	Leaf Scorch
18		Bacterial Spot
19		Early Blight
20		Late Blight
21		Leaf Mold
22	Tomato	Septoria Leaf Spot
23		Spider Mites (Two Spotted Spider Mites)
24		Target Spot
25		Mosaic Virus
26		Yellow Leaf Curl Virus
27	Apple	
28	Bell Pepper	
29	Blueberry	
30	Cherry	
31	Corn	
32	Healthy Grape	No disease
33	Peach	
34	Potato	
35	Raspberry	
36	Soybean	
37	Strawberry	
38	Tomato	

The proposed approach is based on analysis of visual leaf features in a stepwise manner to construct a classifier and is provided below:

3.1 Background Removal

The task of background removal is crucial in the approach as it may reduce the quality of features extracted from the leaf image. This step involves the removal of background to avoid any potential bias in extracted features and trained framework. Color cast removal is performed by normalizing the gray values of three-color channels separately. Moreover, the task of background removal needs to be automated and free from any human influence to increase usability. The background removal can be performed in two ways: i) pixel clustering, and ii) edge detection. We have transformed RGB image to HSV (Hue, Saturation and Value) color space to easily separate the background from the leaf. Fig. 2 provides the demonstration of true-color image and its channels in RGB and HSV form to demonstrate that it is easier to transform the image to HSV and use Hue layer for background removal rather than performing the task of background removal on the true color image itself.

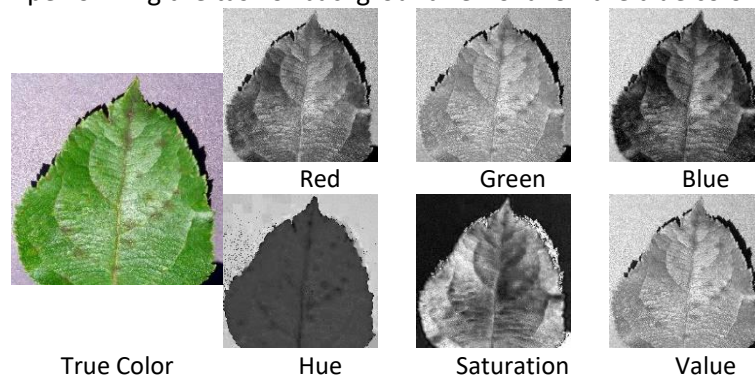


Fig. 2 True Color image (Left) red, green, blue as well as hue, saturation and value channels of the image. The true color image and the color channels have shadows and light reflection which make them less suitable for background removal. This true-color image is transformed to HSV color space and its three layers along with three RGB layers are displayed in Fig. 2. It is visible that Hue image is a better

candidate for segmentation as it cast out all the intensity-related information. Edge preserving blurring is applied on the Hue image using the bilateral filtering and then Watershed algorithm is applied [29]. It finds watershed ridges in an image by treating light pixels as high elevations and dark pixels as low elevations. Eight neighborhood principle is used to segment adjoining regions. At the final stage, small pixel groups are removed by morphological processing which improved the mask. The stages of preprocessing and segmentation are provided in Fig. 3.

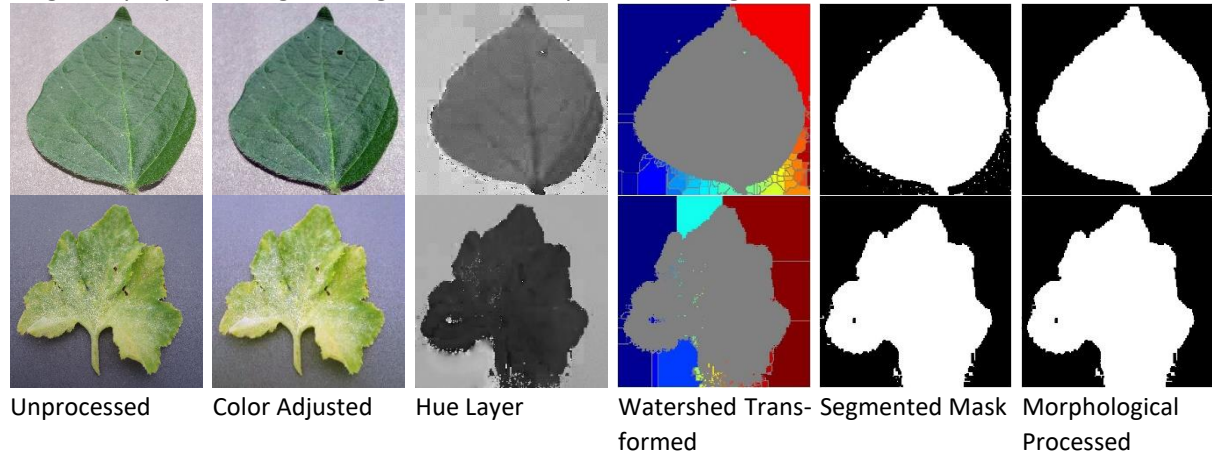


Fig. 3 Steps of Preprocessing and background removal on two sample images

Most of the images are perfectly segmented through this approach and few images had some extra area labeled as the leaf. The segmentation results were slightly poor for Powdery Mildew of Squash shown lower in Fig. 3. It is observed experimentally that slightly poor segmentation performance do not drastically affect the disease classification.

The background removal results of some sample leaf images are provided in Fig. 4. It is important to note that most of the leaf images are perfectly segmented but few of the images are imperfectly segmented containing non-leaf area. Leftmost two images are the segmentation results of the leaf images of Fig. 3. The lower-left image is of squash affected by powdery mildew and the images in this class are particularly poor for background removal.



Fig. 4 Sample Images after background removal

3.2 Healthy and Diseased Classification

In PlantVillage dataset there were two categories of leaves: diseased and healthy. Disease identification involves clustering of the leaves in several regions and segmentation of diseased image region.

This segmented region has been processed to obtain texture and color-based features for disease identification. The healthy leaves do not have any diseased region and therefore, the clustering and segmentation phase results into the extraction of inconsistent leaf region and become the cause of degradation in the identification of disease. The task of classification is separated into two steps: i) categorization into diseased and healthy leaves and ii) identification of plant disease among the 26 disease categories. The flow chart of this approach is shown in Fig. 5.

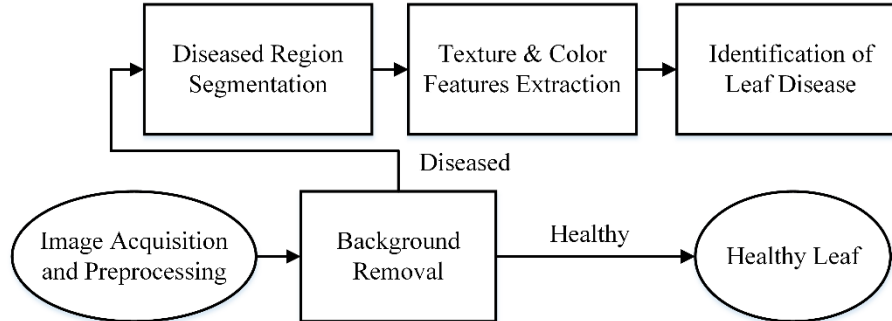


Fig. 5 Flow chart of leaf disease identification

Classification of leaf images into healthy or diseased leaves is done by the bag-of-features approach. Speeded Up Robust Features (SURF) are extracted from images which is a local feature detector and descriptor. The first step of SURF detects interest points through blob detection. Blob is the region of image having points that are in some sense similar to each other and differ from its neighboring regions in terms of color, contrast, or brightness. These interest points can be of different scales and the feature descriptors provide a robust description of the intensity distribution of the pixels within the neighborhood of the interesting point. These features are used to construct a bag-of-features (inspired by the bag of words approach of natural language processing) and 50% of the images (27,155) are used for bag-of-features construction. The number of features is reduced through quantization of feature space using k-means clustering algorithm and the codebook size is kept small for portability to mobile devices. Moreover, 70% of the strongest features from each class (healthy or diseased) are used. The bag-of-features are used to train an SVM model having a linear kernel with two classes to categorize the images as diseased or healthy.

3.3 Segmentation of Diseased Region

Segmentation of diseased regions is useful to extract discriminative features. In this study, the diseased region is extracted for texture feature calculation as it contains the most discriminative portion of the plant disease. Therefore, extraction of diseased segments is a requirement that is described further. Among few options for diseased region segmentation, we have opted for Otsu's algorithm [30] which is computationally inexpensive and a reasonable choice overall. Otsu's algorithm is a simpler and quicker approach to segment the diseased region of the leaf. It works by calculating the threshold in the gray-level histograms. It is assumed that the foreground and background pixels belong to two different Gaussian distributions with different mean and variance values. It finds the ridge between two peaks by maximizing the variance between the two classes and the final point in the gray-level histogram is used as a threshold to segment the image.

The diseased region segmentation is not exceptionally accurate however it is satisfactory. Fig. 6 demonstrates the results of segmentation for Corn affected by common rust, Grape affected by leaf blight, and potato affected by late blight.

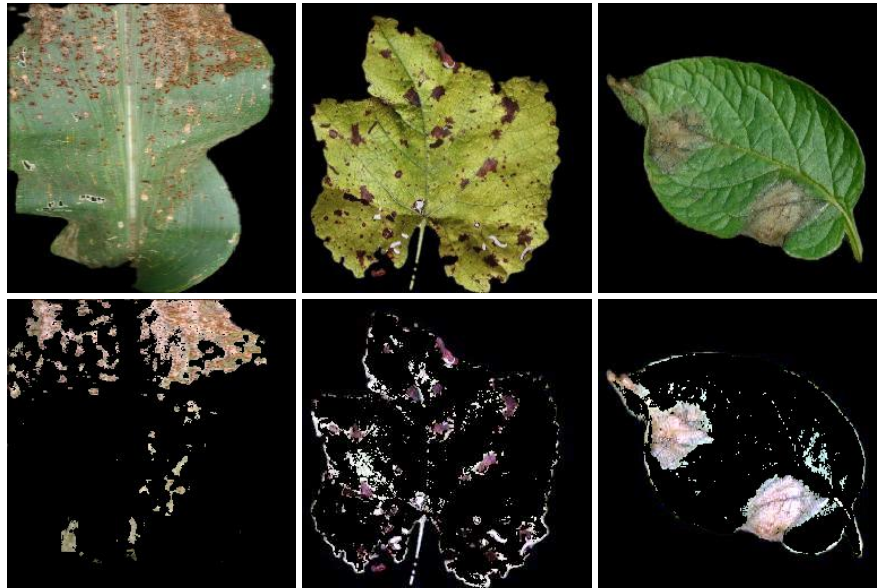


Fig. 6 (Row-1) Diseased Leaf Images of Corn, Grape and Potato, (Row-2) Segmented diseased region of leaves

3.4 Feature Extraction

Feature extraction is an important step and the accuracy of a classification algorithm depends heavily on the feature set. Feature extraction is dimensionality reduction which is done to effectively represent the interesting parts of a diseased region in a compact form. Shape, color and texture-based features represent distinct leaf-based classification features [31, 32]. The shape features can be used in the identification of healthy or diseased leaves but in disease identification, the segmented diseased region has different inter and intra-class variabilities and would be of less use. The color of the diseased region is distinct from the healthy region and stays similar in intra-class samples and vary in inter-class samples. Similarly, the texture of the diseased region depends heavily on the type of disease and can be used as a major predictor. The parameters related to the calculation of these two sets of features are discussed below:

Color Features

There are two main types of color features namely: color histograms and color moments. In the present scenario, color is of lesser importance as a predictor of disease categories. Color histograms provide a larger feature set which is not greatly required in this scenario. Color moments are used to characterize color information which acts as a compact way to represent color information.

Mean: Mean value of the two-dimensional image matrix can be calculated to represent the first color moments. The color image is separated into its three RGB layers and the mean value is calculated for each layer.

Standard Deviation: Second color moment is the standard deviation which represents the distribution of color information around the mean. It is calculated for three RGB layers for a two-dimensional image matrix.

There is a total of six features representing color information in the diseased image region, three for mean and three for standard deviation of red, green and blue layers of RGB image.

Texture Features

The texture is a more important part of disease identification as it represents more information related to the diseased region. Here 22 texture features have been extracted from Gray-Level Co-occurrence Matrix (GLCM) of grayscale leaf image as the color information is already encoded using color moments. The GLCM is calculated over grayscale image for which range is resampled into eight gray-levels. It calculates how frequently a pixel with a gray-level is situated adjacent to a pixel with the value j . The elements (i, j) of the GLCM matrix of 8×8 represent the number of times that pixel with value i occurred adjacent to a pixel with value j .

For an image with 8 different gray-level intensities, the GLCM G of dimension 8×8 is defined over $m \times n$ image I , with an offset parameter (Δ_x, Δ_y) formula in eq. 1 describes it.

$$G_{\Delta_x, \Delta_y}(i, j) = \sum_{x=1}^m \sum_{y=1}^n \begin{cases} 1, & \text{if } I(x, y) = i \text{ and } I(x + \Delta_x, y + \Delta_y) = j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where: i & j are the pixel values; x and y are the spatial positions in the image matrix I ; (Δ_x, Δ_y) are the offset which defines the spatial relation of the matrix and $I(x, y)$ indicates the pixel values at a location (x, y) .

The details of these texture features are provided as follows:

No.	Feature	Formula
1	Uniformity (angular second moment) [33-36]	$\text{Uniformity} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)^2$
2	Entropy [33-36]	$\text{Entropy} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log[p(i, j)]$
3	Contrast [33-36]	$\text{Contrast} = \sum_{i=1}^{N_g} (i - j)^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\}$
4	Dissimilarity [33, 34, 36]	$\text{Dissimilarity} = \sum_{i=1}^{N_g} (i - j) \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\}$
5	Inverse Difference Moment (Homogeneity) [33, 36]	$\text{Homogeneity} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i, j)}{1 + (i - j)^2}$
6	Inverse Difference (ID) [33]	$\text{ID} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i, j)}{1 + i - j }$
7	Correlation [36]	$\text{Correlation} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{ijp(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
8	Autocorrelation [34]	$\text{Autocorrelation} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \bar{p}(i - k, j - k)$ Where k is the amount of shift
9	Cluster Shade (CS) [33, 34]	$\text{CS} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i + j - \mu_x - \mu_y)^3 \bar{p}(i, j)$
10	Cluster Prominence (CP) [33, 34]	$\text{CP} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i + j - \mu_x - \mu_y)^4 \bar{p}(i, j)$
11	Maximum Probability (MP) [33, 34]	$\text{MP} = \max_{i, j} p(i, j)$
12	Sum of Squares (SS) [35]	$\text{SS} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i - \mu)^2 p(i, j)$
13	Sum Average (SA) [35, 36]	$\text{SA} = \sum_{i=2}^{2N_g} ip_{x+y}(i)$

14	Sum Variance (SV) [35, 36]	$SV = \sum_{i=2}^{2N_g} \left(i - \left[\sum_{i=2}^{2N_g} p_{x+y}(i) \right] \right)^2$
15	Sum Entropy (SE) [35, 36]	$SE = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$
16	Difference Variance (DV) [35, 36]	$DV = \sum_{i=2}^{2N_g} \left(i - \left[\sum_{i=2}^{2N_g} i p_{x-y}(i) \right] \right)^2$
17	Difference Entropy (DE) [35, 36]	$DE = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$
18	Information Measures of Correlation (IMC) [35]	$IMC_1 = \frac{HXY - HXY_1}{\max\{HX, HY\}}$ $IMC_2 = \sqrt{1 - e^{[-2.0(HXY_2 - HXY)]}}$ <p>Where; $HXY = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p(i, j))$</p>
19	Maximal Correlation Coefficient (MCC) [35]	$MCC = \sqrt{F \left(\sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)} \right)}$ <p>Where; F is the second-largest Eigenvalue</p>
20	Inverse Difference Normalized (IDN) [33]	$IDN = \sum_{i,j=1}^G \frac{C_{ij}}{1 + i - j ^2 / G^2}$
21	Inverse Difference Moment Normalized (IDMN) [33]	$IDMN = \sum_{i,j=1}^G \frac{C_{ij}}{1 + (i - j)^2 / G^2}$

There is a total of 22 texture features which represent the texture information of the segmented diseased region of leaf image.

3.5 Feature Scaling

It is a method to scale the range of values of a feature vector. It is also known as feature normalization or standardization. Feature values are computed in different units or may represent different parameters so their ranges vary widely. It is to note that many classifiers use some distance measure, such as Euclidian distance, between the points. If one feature has a wide range of values, the distance measure will be largely affected by this specific feature, so, the range of all the features should be normalized to reflect proportionality in the distance measure. Moreover, the convergence time of gradient descent is less with feature scaling and it is sometimes less for stochastic gradient descent as well. In SVM, feature scaling can reduce the time to find support vectors and affect the results as well. The performance of K-NN, SVM, logistic regression, perceptron, artificial neural networks and linear discriminant analysis is largely affected by feature scaling.

Different feature scaling methods are used in the literature and the selection of a particular method depends on the requirements. Feature scaling usually is applied to change the range of features to a constrained value such as [0,255] in case of images or [0, 1] or [-1, 1] in the case of features with real values. Max-min normalization is the most widely used method in which features are scaled to a specified range i.e. [0, 1] by scaling their maximum and minimum values to these points. In this work, standardization is used which scale the features such that they have zero mean and unit variance. This is a widely used scaling method for SVM, ANN and linear regression [37]. The formula in eq. 2 is used for standardization of features:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (2)$$

3.6 Feature Selection

Feature selection is employed to select a subset of features for model construction. There are four major reasons to perform feature selection: i) simplification of model, ii) shorter training time, iii) avoidance from curse of dimensionality [38] and (iv) reduction of overfitting. The basic idea behind feature selection is that the feature set can contain irrelevant as well as redundant features. Redundant and irrelevant are different terms, as one relevant feature may be redundant in the presence of some other correlated feature.

Texture and color features have a lot of dependencies on each other and have a high correlation in some cases. Inclusion of all the features may cause degradation in performance and will result in slower training. Therefore, feature selection is crucial. There are different approaches for feature selection and we have explored two of them for our problem: i) sequential feature selection and ii) ReliefF algorithm. Sequential feature selection can be used in two configurations, one is forward and the other is backward. The Forward Feature Selection (FFS) starts from one feature and iteratively adds features and compares the performance based on predefined criteria. The features are added till the accuracy stop increasing or starts decreasing. In backward feature elimination, the process is started with all the features and one feature is removed at a time and performance is compared. This work has used forward feature selection in which the training algorithm is trained and cross-validated. We have used SVM with 10-fold Cross-Validation (CV) for evaluation and the feature with maximum accuracy are retained. The process of feature inclusion is continued till there is no increase in accuracy. Then 19 features have been selected in the given order and achieved a CV accuracy of 0.934 at the end, Table 2 provides the list of these features and the CV accuracy.

Table 2: Feature selected through forward feature selection with Multi-SVM as criterion

Sr. No.	Feature Name	Cross-Validation Accuracy
1	Correlation	0.704019
2	Autocorrelation	0.718989
3	Sum of Squares	0.728080
4	Standard Deviation of Blue	0.760980
5	Maximal correlation coefficient	0.768141
6	Information measures of correlation (2)	0.770524
7	Mean of Blue	0.775902
8	Sum Variance	0.783225
9	Sum Entropy	0.802779
10	Cluster Prominence	0.802957
11	Standard Deviation of Red	0.821492
12	Uniformity	0.831278
13	Contrast	0.851472
14	Difference entropy	0.859829
15	Homogeneity	0.880204
16	Entropy	0.908171
17	Mean of Green	0.914150
18	Inverse difference	0.930732
19	Inverse difference moment normalized	0.934014

ReliefF algorithm is a feature ranking algorithm and this ranking can be truncated based on weights to select a subset of features. It finds weights of the features in the cases where the target-dependent variable is a multiclass categorical variable. The algorithm rewards the features that give different values to neighbors of different classes and penalize the features that give a different value to neighbors of same class. In initialization, all the feature weights are set to zero and then a random observation is iteratively selected. K-nearest neighbor's algorithm is used to select the nearest observations to randomly selected observations and updates the weights for all the classes for that feature. The algorithm provides two measures as an output, feature ranks and weights. The ranking is an integer array with an index number of each feature ordered in decreasing rank. The weights array contains the prediction weights for all the ranked vectors. Negative weights indicate that the feature

may do more harm than good in classification performance. Eight features out of 29 have positive weight values and rest of the 20 features have negative or zero weight values. We have discarded the features with negative weights and selected the ones having positive weight values. These eight selected features along with their rank and weight are provided in Table 3.

The two feature subsets, one with eight features and the other with nineteen features were used to check their classification performance and it was observed that the features selected with FFS are better suited for classifier training.

Table 3: Selected features along with their rank and weights

Rank	Feature Name	Feature Weight
1	Standard Deviation of Blue	0.0015
2	Mean of Blue	0.0010
3	Standard Deviation of Red	0.0008
4	Standard Deviation of Green	0.0007
5	Mean of Red	0.0007
6	Inverse difference	0.0002
7	Entropy	0.0002
8	Difference entropy	0.0002

3.7 Classifier Selection

The final stage of the work is selection of a suitable classification algorithm for classification of leaf disease to the category they belong. We have chosen five major classification algorithms to check their suitability for classification. The algorithm with best performance is optimized for its hyper-parameters to form a final model. This study has used multi-SVM, K-NN, Naïve Bayes, Random Forest and Artificial Neural Networks (ANN).

Multi-SVM: The SVM is a supervised learning algorithm that is used for classification, regression, and clustering or outlier detection. SVM is a binary linear classifier that separates two classes using maximum margin hyperplane. A good separation is provided by a hyperplane that has the largest distance to the nearest data points, therefore referred to as maximum margin hyperplane, as larger the margin lower the generalization error. The feature set may exist in a finite-dimensional space but it is not linearly separable in that space. To make the data linearly separable it is transformed to a higher dimensional space using a kernel function where it is linearly separable.

The binary classification nature of SVM is overcome by several methods one-vs-all and one-vs-one are two of them. In one-vs-all multi-SVM, all the classes are selected separately and one is taken as positive labels and all other classes as negative labels and a total of N classifiers are trained. The problem with imbalanced data is that one class has fewer samples than all the other classes and general SVM implementation may behave poorly. One-vs-one train separate classifier for each class paired with all others, one by one, and thus overcoming the problem of imbalance. This leads to $\frac{N(N-1)}{2}$ classifiers making it much more computationally expensive.

The present study has used the one-vs-one multi-SVM and tried linear, quadratic, cubic and Gaussian kernels whereas cubic SVM provided best cross-validation performance. The kernel functions used for SVM are provided in Table 4.

K-NN: It is a non-parametric algorithm that is used for classification, regression or outlier detection. It is a lazy learning algorithm that doesn't try to construct a model, rather simply stores the training examples of the data. The classification is performed through simple majority voting of the k-nearest neighbors. This is a preferred algorithm for noisy or large training data and is easier to implement. The problem with K-NN is selection of the value of K. Smaller value of K makes finer decision boundary resulting in overfitting, and larger K value results in smoother boundary resulting in poor classification accuracy due to higher bias. Determination of suitable value of K is computationally expensive as it needs to compute distance of each example with all training samples.

Table 4: Kernel functions used with SVM

Sr. No.	Kernel Method	Kernel Function
1	Linear	$K(x, y) = \text{sum}(x * y)$
2	Polynomial	$K(x, y) = 1 + \text{sum}(x * y)^d$
3	Gaussian	$K(x, y) = e^{-\left(\frac{\ x-y\ ^2}{2\sigma^2}\right)}$

Naïve Bayes: It is a probabilistic classification algorithm based on Bayes theory having an assumption that each feature is independent of others. It has a wrong and simplistic assumption that each feature is independent of the presence of any other feature in the feature set. This assumption also helps to alleviate the problems arising due to the curse of dimensionality. However, Naïve Bayes has good performance in real-world situations, it requires small amount of training data and is fast to compute.

Naïve Bayes is not a single algorithm but a set of algorithms for classification which are based on common assumption that a specific feature is independent of any other feature. It can be trained efficiently using supervised learning approach and in many settings, parameter estimations are based on maximum likelihood.

Random Forest: Overfitting is the inherent problem of decision trees that is overcome using different bagging or boosting approaches. The problem of overfitting is addressed by random forest which fits many decision trees on several sub-samples of the training dataset and uses an average for improvement of the predictive accuracy. The sub-sample size is always same as the original sample size except the samples are drawn with replacement. The principle advantage of the reduction of overfitting makes random forest more accurate than simple decision trees in most cases. However, it is a complex algorithm and its prediction speed is slow.

Artificial neural networks: ANNs are computational algorithms that are roughly inspired by biological artificial neural networks. These systems learn to perform tasks from labelled examples, without any task specific rules. For a classification task, they will automatically adjust the weight of their neurons based on training examples and try to classify new instances based on learned weights. A multi-layer artificial neural network has an input layer, an output layer and one or more hidden layers. Normally, the number of neurons in the input layer are equal to feature vector length and number of neurons in the output layer are equal to the number of classes. The number of hidden layers and the number of neurons in each hidden layer depends on the particular problem set and can't be computed beforehand.

In this study, we have 29 neurons in input layers, 26 neurons in the output layer and number of neurons in hidden layer are selected through search.

4. Results and Discussions

Existing studies have used feature-based or CNN-based classification to categorize the leaves of PlantVillage dataset into one of 38 categories. The proposed approach mainly relied on texture features extracted through statistics of GLCM applied on segmented diseased region of the leaf. As the segmentation stage of diseased leaf area is automated, when applied on healthy leaves it extracts some leaf area and confuses the classifier. The confusion matrix of the diseased or healthy plant classification stage is provided in Table 5:

Table 5 Confusion matrix of healthy and diseased plant classes

	Healthy	Diseased
Healthy	99.7848	0.2152
Diseased	0.8280	99.1720

Precision, recall, accuracy and F1-Score of the healthy and diseased plant classification are provided in Table 5. The results of the classification are reasonable but existing approaches have not used classification between healthy and diseased leaves for this dataset.

Table 6 Evaluation results of diseased and healthy leaf classification

Precision	Recall	Accuracy	F1-Score
-----------	--------	----------	----------

SVM	99.78	99.18	99.48	99.48
-----	-------	-------	-------	-------

The second phase involves the segmentation of diseased leaf region of 39,226 images. Six color and twenty-two texture features were extracted and a feature vector with a length of 29 features is formed. This feature vector is standardized so that the feature vector has zero mean and unit variance. Standardization is applied to remove the feature range effect on distance-based classifiers, reduce model complexity and improve the gradient calculation. Feature selection using ReliefF algorithm and forward feature selection using Multi-SVM are used for feature subset selection. Here, eight features have been selected with ReliefF algorithm having positive weight values and nineteen features using FFS which provided improvement in accuracy. The evaluation of three feature sets, i) complete feature set, ii) sub-set with ReliefF algorithm and iii) sub-set with FFS is performed on five classifiers discussed in section IV (G).

It has been observed that the classification accuracy of subset selected using ReliefF algorithm is significantly less than the classification accuracy of the complete feature set. Whereas, sub-set with FFS provided slightly better classification accuracy than complete feature set. It has been a favorable decision which provides better performance in terms of time and classification accuracy. The results of classification evaluation with three feature sets are provided in Table 7.

Table 7 Classifier evaluation results with complete feature set and two subsets

Method	Complete 29 features	ReliefF 8 features	FFS 19 features
Multi-SVM	92.1	80.2	93.4
K-NN	84.5	85.5	86.7
Naïve Bayes	78.4	76.1	82.4
Random Forest	82.2	77.2	87.9
ANN	91.7	87.4	92.6

The feature subset selected after feature selection stage is used for fine-tuning the Multi-SVM and ANN as they both provided highest accuracy in initial stage.

The ANN was configured with one hidden layer having neurons varying from 10 to 100 neurons and two hidden layers with neurons varying from 10 to 50 neurons and evaluated based on Cross-Entropy (CE). The network with the least cross-entropy has been tested for classification accuracy of which turned out to be 93.4% with the neural net of Figure 7.

Table 8 Artificial neural network performance with different number of neurons

Network Size	CE	Network Size	CE
<i>One Hidden Layer</i>		<i>Two Hidden Layers</i>	
10	0.0476	10:10	0.0489
20	0.0378	15:10	0.0454
30	0.0342	20:10	0.0429
40	0.0332	20:15	0.0376
50	0.0328	30:20	0.0352
60	0.0351	40:30	0.0340
70	0.0336	40:40	0.0325
80	0.0335	50:30	0.0332
90	0.0326	50:40	0.0327
100	0.0383	50:50	0.0313

Table 8 provides the network performance with different numbers of neurons. In neural networks with one hidden layer, minimum cross-entropy of 0.0328 is achieved with 50 neurons in the hidden layer. For second neural net with two hidden layers, minimum cross-entropy of 0.0313 is achieved with 50 neurons in the first hidden layer and 50 neurons in the second hidden layer.

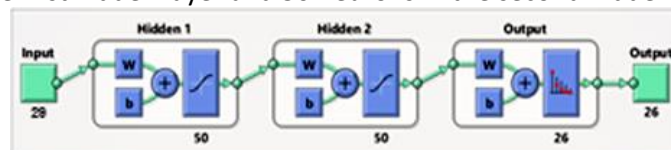


Fig. 7 Artificial neural network architecture with two hidden layers

The multi-SVM with one-vs-one approach was used and cross-validation accuracy was compared for linear, quadratic, cubic and Gaussian kernels. The best CV accuracy is provided by multi-SVM with cubic kernel having performance parameters provided as in Table 9. Note that the accuracy provided by SVM with cubic kernel is higher than the accuracy provided by the artificial neural network with 50 neurons in the first hidden layer and 50 neurons in the second hidden layer.

Table 9 Performance parameters for SVM with different kernels, the time is taken in seconds and prediction speed is taken in observations per seconds

	Linear	Quadratic	Cubic	Gaussian
Accuracy	84.39%	95.19%	98.99%	83.89%
Training Time	137s	645s	3177s	107s
Prediction Speed	3800 obs/s	1300 obs/s	1800 obs/s	510 obs/s

It can be observed that SVM has provided impressive performance with cubic kernel but it has taken longer training time (almost 53 minutes). The prediction speed for SVM with cubic kernel, however, is only slower than linear kernel. Since training is a one-time job and can be done on a faster machine, prediction speed is the desired parameter for its practical use. Therefore, SVM with cubic kernel was selected as the final model. The confusion matrix for the final model is provided in the Table 10 below:

Table 10 Plant disease classification evaluation results for Disease Classification

	Precision	Recall	Accuracy	F1-Score
Proposed	99.35	99.31	99.31	99.33

Table 11 provides the results of classification evaluation for the task of plant disease identification. Please note that precision is macro-average precision which is averaged across 26 classes. The recall is also average of the recall for each class. The proposed model has provided an accuracy of 99.31 and F1-score of 99.33 which is quite high and comparable to existing approaches. The high accuracy can be explained based on rich set of texture and color features and with the use of feature selection which provides subset of most discriminatory features.

It is to be noted that the results of Table 10 are for 26 diseased classes only, whereas the complete dataset contains 38 classes. In the literature, only the leaf category is classified so the results of Table 10 can't be compared with them. However, we have calculated the aggregate classification performance by accumulating the classification scores of both stages which are provided in Table 12. It is to be noted that final classification score is slightly less than the disease classification module performance provided in Table 11.

Table 11 Plant disease classification evaluation for Final Model

	Precision	Recall	Accuracy	F1-Score
Proposed	99.13	98.49	98.79	98.81

The difference in classifier performance is due to their limitations such as Naïve Bayes performs poorly due to lack of all posterior probabilities, random forest can't perform well due to limited performance of decision tree and their averaging over the number of trees can give intermediate accuracy. K-NN can perform well in cases where features can be discriminated against based on some distance metric and the artificial neural networks has provided comparable accuracy to SVM and the slightly lower accuracy is due to difficulty in finding the global minima. The artificial neural network has been trained numerous times and there is slight change in its accuracy which is due to the initialization of its weight and difficulty to find a global minimum. The ANN is iterated several times with different numbers of neurons in one and two hidden layers. The iterations are limited due to computational complexity and can be repeated more times if a good solution is not achieved using other classifiers.

Table 12 Disease Classification Confusion Matrix

[illegible]

Table 13 provides a comparison of the proposed method with existing work. Most of the methods has used train/test split for training and validation of their method whereas some methods have employed Cross-Validation (CV) in which the data is divided into $\frac{1}{n}$ folds and during each iteration $n - 1$ folds are used for training and one of them is used for testing. After completion of n-folds, the average values are obtained.

Table 13 Comparison with existing methods

Year	Existing Work	Accuracy	Precision	Recall	F1-Score	Remarks
2016	Mohanty et al. [14]	99.35	99.35	99.35	99.34	38 classes with 60/40 split
2017	Islam et al. [39]	95	95	95	95	3 classes with 60/40 split
2017	Hlaing et al. [40]	78.7	-	-	-	38 classes with 5-fold CV
2017	Hlaing Et al. [41]	84.7	-	-	-	7 classes with 10-fold CV
2017	Amara et al. [42]	99.72	99.70	99.72	99.71	3 classes with 50/50 split
2017	Durmuú et al. [43]	95.65	-	-	-	10 classes with 80/20 split
2017	Brahimi et al. [44]	99.185	98.529	98.532	98.518	9 classes with 5-fold CV
2017	Yamamoto et al. [45]	92	90	92	91	9 classes with 80/20 split
2018	Yuan et al. [27]	95.97	-	-	-	38 classes with 80/20 split
2018	Goncharov et al. [16]	99	-	-	-	5 classes with 75/25 split
2018	Wang et al. [46]	90.84	-	-	-	8 classes with 80/20 split
2018	Zhang et al. [28]	91.79	-	-	-	38 classes with 80/20 split
2018	Ferentinos et al. [12]	99.53	-	-	-	58 classes with 80/20 split
2018	Yadav et al. [47]	97.39	97.76	97.39	97.34	23 classes with 70/30 split
2019	Kour et al. [48]	95.23	-	-	-	7 classes with 70/15/15 split
2018	Pardede et al. [49]	87.01	-	-	-	7 classes with 80/20 split
2018	Suryawati et al. [50]	95.24	-	-	-	10 classes with 80/10/10 split
2019	Khandelwal et al. [13]	99.37	-	-	-	57 classes with 80/20 split
2019	Barbedo et al. [11]	94	-	-	-	14 classes with 80/20 split
2019	Baranwal et al. [51]	98.54	-	-	-	38 classes with 80/20 split
	Proposed	99.13±0.57	98.49±0.59	98.79 ±0.57	98.81 ±0.58	38 classes with 10-fold CV

It can be noted from Table 13 that 7 existing works have claimed higher accuracy than the proposed scheme but four of them have used subset of the PlantVillage dataset and targeted a specific plant for disease identification. However, three of them have used similar or higher number of classes for disease identification. The three schemes with higher accuracies have used deep learning-based methods which employ image augmentation to increase the training dataset size. These methods take a longer time to train and require more computational and storage space to perform prediction which is a limitation for mobile devices as the ultimate utilization of the proposed scheme is deployment on mobile devices. Moreover, the testing scheme used in our case is 10-fold cross-validation which is a more rigorous method to check the behavior of classification performance as compared to hold-out validation such as 80/20 or 60/40. We have provided the standard deviation of values along with the performance metrics which are obtained over 10 train test splits. Moreover, the proposed algorithm work in a two-stage configuration and direct comparison of the proposed approach with other methods is not possible. There are 12 healthy leaf classes and 26 diseased leaf classes and we have performed the classification of diseased leaf only in the second stage. Further classification of the healthy leaf images is not in the scope of

this work as we have targeted only to identify if a leaf image is healthy or diseased and then identify the category of disease if it is a diseased leaf image.

After finalizing the proposed model and evaluation of its results on PlantVillage dataset which is collected in a controlled environment, it is important to test available model on data collected independently for generalization. The images from online sources and some small datasets are gathered in self-collected dataset having same classes as in PlantVillage. Some images have been collected from the agricultural extension services and a set of 227 images is gathered. These images are selected based on their resemblance with the images in the database as more the variability in training and testing database, the more is the gap between their performances. The proposed model is tested on this dataset and classification accuracy of 91.40% is achieved for healthy and unhealthy classification. In the case of disease category classification, the accuracy is 82.47%. Note that a decrease in classification accuracy of plant disease is due to difficulty in segmentation of the diseased leaf area. An improvement in segmentation of diseased leaf areas can result in even better disease identification accuracy and provide more reliable solution. Some of the schemes described in Table 13 have cross-validated their approach on an independently collected database and reported decrease in the classification accuracy. The proposed approach in comparison to deep learning-based approaches requires low memory and less computation thus making it feasible for deployment on mobile devices.

Conclusion

In this study, a model for identification of plant disease based on leaf image is proposed. The proposed model works in two stages, one categorizes the leaf image either as healthy or diseased. The second stage identifies the disease affecting the leaf in case of unhealthy leaf only. The first stage uses a bag-of-features approach and provided a recall of 99.48%. The second stage involves extraction of diseased leaf area and calculates a set of 29 texture and color features. Feature standardization is used to reduce the bias due to feature range and feature selection and is performed to select the most discriminatory features. An ensemble of classifiers involving SVM, K-NN, Naïve Bayes, random forest and artificial neural networks was trained on the selected feature set and the classifier with best performance is selected. The SVM provided best performance with polynomial kernel of degree three and final model is constructed. The classification accuracy of final model for disease classification is 98.79% with a standard deviation of 0.57 over 10-fold cross-validation. The proposed model's ability to generalize is evaluated by collecting a dataset of 227 images and using the proposed model for evaluation. The classification accuracy of diseased and healthy plant identification is 91.40% and for disease, category identification is 82.47%. The proposed model is a prototype system and it can be used to train on different disease classification scenarios. The only limitation is that the system will be able to classify only images that are visually distinguishable and provided in the training database.

Future Work

- a. An integrated solution containing plant species identification using leaf image in the first stage and plant disease identification in second stage can be formulated to improve the usability of the proposed solution.
- b. Specialized models for high-value crops or specific crops with more diseases can be designed to identify the disease and categorize its severity based on fine-grained recognition.

Declarations

Funding: There are no funding sources for this project.

Conflicts of Interest: It is declared that there are no financial and other competing conflicts of interests.

Availability of data and material: The data used for experiments is available at <https://github.com/spMohanty/PlantVillage-Dataset>

Code availability: The source code of the project will be made public after acceptance of the manuscript.

Authors' contributions: The contributions of the authors are listed below:

N. Ahmed: Conception, coding, experimentations and writeup.

H. M. S. Asif: Conception, supervision and review.

G. Saleem: Coding, experimentations, writeup and review.

M. U. Younus: experimentations, writeup and review.

Sadia Anwar: Proof Read and reviewed the manuscript.

Muhammad Rizwan Anjum: Proof Read and reviewed the manuscript.

References

1. Savary, S., et al., *Crop losses due to diseases and their implications for global food production losses and food security*, 2012, Springer.
2. Pertot, I., et al., *Identificator: A web-based tool for visual plant disease identification, a proof of concept with a case study on strawberry*. Computers and electronics in agriculture, 2012. **84**: p. 144-154.
3. Colwell, R., *Determining the prevalence of certain cereal crop diseases by means of aerial photography*. Hilgardia, 1956. **26**(5): p. 223-286.
4. Mahlein, A.-K., *Plant disease detection by imaging sensors—parallels and specific demands for precision agriculture and plant phenotyping*. Plant disease, 2016. **100**(2): p. 241-251.
5. Bock, C.H. and F.W. Nutter Jr, *Detection and measurement of plant disease symptoms using visible-wavelength photography and image analysis*. Plant Sciences Reviews 2011, 2012: p. 73.
6. Bock, C., et al., *Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging*. Critical Reviews in Plant Sciences, 2010. **29**(2): p. 59-107.
7. Savary, S., et al., *Quantification and modeling of crop losses: a review of purposes*. Annu. Rev. Phytopathol., 2006. **44**: p. 89-112.

8. Dhingra, G., V. Kumar, and H.D. Joshi, *Study of digital image processing techniques for leaf disease detection and classification*. Multimedia Tools and Applications, 2017: p. 1-50.
9. Dhingra, G., V. Kumar, and H.D. Joshi, *Study of digital image processing techniques for leaf disease detection and classification*. Multimedia Tools and Applications, 2018. **77**(15): p. 19951-20000.
10. Hughes, D. and M. Salathé, *An open access repository of images on plant health to enable the development of mobile disease diagnostics*. arXiv preprint arXiv:1511.08060, 2015.
11. Barbedo, J.G.A., *Plant disease identification from individual lesions and spots using deep learning*. Biosystems Engineering, 2019. **180**: p. 96-107.
12. Ferentinos, K.P., *Deep learning models for plant disease detection and diagnosis*. Computers and Electronics in Agriculture, 2018. **145**: p. 311-318.
13. Khandelwal, I. and S. Raman, *Analysis of Transfer and Residual Learning for Detecting Plant Diseases Using Images of Leaves*, in *Computational Intelligence: Theories, Applications and Future Directions-Volume II* 2019, Springer. p. 295-306.
14. Mohanty, S.P., D.P. Hughes, and M. Salathé, *Using deep learning for image-based plant disease detection*. Frontiers in plant science, 2016. **7**: p. 1419.
15. Zhang, S., et al., *Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG*. Optik-International Journal for Light and Electron Optics, 2018. **157**: p. 866-872.
16. Goncharov, P., et al., *ARCHITECTURE AND BASIC PRINCIPLES OF THE MULTIFUNCTIONAL PLATFORM FOR PLANT DISEASE DETECTION*. 2018.
17. Li, J., J. Jia, and D. Xu. *Unsupervised Representation Learning of Image-Based Plant Disease with Deep Convolutional Generative Adversarial Networks*. in *2018 37th Chinese Control Conference (CCC)*. 2018. IEEE.
18. Phadikar, S., J. Sil, and A.K. Das, *Rice diseases classification using feature selection and rule generation techniques*. Computers and electronics in agriculture, 2013. **90**: p. 76-85.
19. Baquero, D., et al. *An image retrieval system for tomato disease assessment*. in *Image, Signal Processing and Artificial Vision (STSIVA), 2014 XIX Symposium on*. 2014. IEEE.
20. Patil, J.K. and R. Kumar, *Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features*. Engineering in Agriculture, Environment and Food, 2017. **10**(2): p. 69-78.
21. Sandika, B., et al. *Random forest based classification of diseases in grapes from images captured in uncontrolled environments*. in *Signal Processing (ICSP), 2016 IEEE 13th International Conference on*. 2016. IEEE.
22. Oberti, R., et al., *Automatic detection of powdery mildew on grapevine leaves by image analysis: Optimal view-angle range to increase the sensitivity*. Computers and Electronics in Agriculture, 2014. **104**: p. 1-8.
23. Sharif, M., et al., *Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection*. Computers and Electronics in Agriculture, 2018. **150**: p. 220-234.
24. Singh, V. and A.K. Misra, *Detection of plant leaf diseases using image segmentation and soft computing techniques*. Information Processing in Agriculture, 2017. **4**(1): p. 41-49.
25. Bai, X., et al., *A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images*. Computers and Electronics in Agriculture, 2017. **136**: p. 157-165.
26. Picon, A., et al., *Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild*. Computers and Electronics in Agriculture, 2018.
27. Yuan, Y., S. Fang, and L. Chen. *Crop Disease Image Classification Based on Transfer Learning with DCNNs*. in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. 2018. Springer.
28. Zeng, W., et al. *High-Order Residual Convolutional Neural Network for Robust Crop Disease Recognition*. in *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*. 2018. ACM.
29. Shafarenko, L., M. Petrou, and J. Kittler, *Automatic watershed segmentation of randomly textured color images*. IEEE transactions on Image Processing, 1997. **6**(11): p. 1530-1544.
30. Vala, H.J. and A. Baxi, *A review on Otsu image segmentation algorithm*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2013. **2**(2): p. 387-389.
31. Saleem, G., et al., *Automated analysis of visual leaf shape features for plant classification*. Computers and Electronics in Agriculture, 2019. **157**: p. 270-280.

32. Ahmed, N., U.G. Khan, and S. Asif, *AN AUTOMATIC LEAF BASED PLANT IDENTIFICATION SYSTEM*. Science International, 2016. **28**(1).
33. Clausi, D.A., *An analysis of co-occurrence texture statistics as a function of grey level quantization*. Canadian Journal of remote sensing, 2002. **28**(1): p. 45-62.
34. Soh, L.-K. and C. Tsatsoulis, *Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices*. CSE Journal Articles, 1999: p. 47.
35. Haralick, R.M., K. Shanmugam, and I.H. Dinstein, *Textural features for image classification*. IEEE Transactions on systems, man, and cybernetics, 1973. **3**(6): p. 610-621.
36. Santos, T., et al., *MRI texture analysis reveals bulbar abnormalities in Friedreich ataxia*. American Journal of Neuroradiology, 2015.
37. Grus, J., *Data science from scratch: first principles with python* 2015: "O'Reilly Media, Inc."
38. Liu, H. and L. Yu, *Toward integrating feature selection algorithms for classification and clustering*. IEEE Transactions on knowledge and data engineering, 2005. **17**(4): p. 491-502.
39. Islam, M., et al. *Detection of potato diseases using image segmentation and multiclass support vector machine*. in *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*. 2017. IEEE.
40. Hlaing, C.S. and S.M.M. Zaw. *Plant diseases recognition for smart farming using model-based statistical features*. in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. 2017. IEEE.
41. Hlaing, C.S. and S.M.M. Zaw. *Model-Based Statistical Features for Mobile Phone Image of Tomato Plant Disease Classification*. in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. 2017. IEEE.
42. Amara, J., B. Bouaziz, and A. Algergawy. *A Deep Learning-based Approach for Banana Leaf Diseases Classification*. in *BTW (Workshops)*. 2017.
43. Durmuş, H., E.O. Güneş, and M. Kirci. *Disease detection on the leaves of the tomato plants by using deep learning*. in *2017 6th International Conference on Agro-Geoinformatics*. 2017. IEEE.
44. Brahimi, M., K. Boukhalfa, and A. Moussaoui, *Deep learning for tomato diseases: classification and symptoms visualization*. Applied Artificial Intelligence, 2017. **31**(4): p. 299-315.
45. Yamamoto, K., T. Togami, and N. Yamaguchi, *Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture*. Sensors, 2017. **17**(11): p. 2557.
46. Wang, J., et al. *CNN Transfer Learning for Automatic Image-Based Classification of Crop*. in *Image and Graphics Technologies and Applications: 13th Conference on Image and Graphics Technologies and Applications, IGTA 2018, Beijing, China, April 8–10, 2018, Revised Selected Papers*. 2018. Springer.
47. Yadav, R., Y.K. Rana, and S. Nagpal. *Plant Leaf Disease Detection and Classification Using Particle Swarm Optimization*. in *International Conference on Machine Learning for Networking*. 2018. Springer.
48. Kour, V.P. and S. Arora, *Particle Swarm Optimization based Support Vector Machine (P-SVM) for the Segmentation and Classification of Plants*. IEEE Access, 2019.
49. Pardede, H.F., et al. *Unsupervised Convolutional Autoencoder-Based Feature Learning for Automatic Detection of Plant Diseases*. in *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. 2018. IEEE.
50. Suryawati, E., et al. *Deep Structured Convolutional Neural Network for Tomato Diseases Detection*. in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. 2018. IEEE.
51. Baranwal, S., S. Khandelwal, and A. Arora, *Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection*. Available at SSRN 3351641, 2019.



Nisar Ahmed his PhD and M.Sc. Computer Engineering from the department of computer engineering, University of Engineering and Technology, Lahore,

Pakistan. His areas of interest include Image Quality Assessment, Multimedia Systems and Computer Vision.



Hafiz Muhammad Shahzad Asif obtained his Ph.D. degree in informatics from the University of Edinburgh, UK in 2012. He is working as Chairman and Associate Professor at the Department of Computer Science, University of Engineering and Technology, Lahore (New Campus), Pakistan.



Gulshan Saleem is a doctoral student at the department of computer science, COMSATS University Islamabad, Lahore Campus. She has done her MS software engineering from College of Electrical and Mechanical Engineering, National University of Science and Technology, Rawalpindi and her BS software engineering from Fatima Jinnah Women University, Rawalpindi. Her areas of interest include machine learning, computer vision and digital image processing.



Dr. Engr. Muhammad Usman Younus received his Doctorate and master's degree in engineering from the University of Toulouse (III) Paul Sabatier France and University of Engineering and Technology Lahore Pakistan 2020 & 2014, respectively. His research interests include Wireless Sensor Network, Software Defined Networking, Energy Optimization, Wireless Communication, and Machine Learning. He has more than twenty international journal and conference publications. He is a member of PEC, IEEE, etc. and reviewer of more than ten journals and conferences.



Sadia Anwar received her degree in Doctor of Pharmacy in 2011 from Government College University, Pakistan. She had worked for three years as a community pharmacist. She also served as a Hospital Pharmacist and Drug information consultant. She arrived in Denmark, December 2015 and started working as a Guest Researcher at CTiF in the department of electronic systems, Aalborg University under the supervision of Professor Ramjee Prasad. She served for Interdisciplinary area specifically more focused in 4 sections: Medicine, Telecommunication, Big data and economics. She joined

Aarhus University in December 2016 in Department of Business Development and Technology as Research Scientist for seven months. Now, she is working as a Research Assistant in the same department from October 2017. Her research interests are based on the interdisciplinary area of Medicine, eHealth, ICT, and Business Development. She is also Working with an

EU project vCare (Virtual Coaching Activities for Rehabilitation in Elderly)—EU Horizon 2020 Research and Innovation funding programme. She is also involved in administrative and managerial tasks. She has produced many research articles in the conferences and Journals.



Dr. Engr. Muhammad Rizwan Anjum received his Ph.D degree from Beijing Institute of Technology, Beijing China in 2015. M. Engg. in Telecommunication and Control Engineering and B.Engg. in Electronic Engineering in 2011 & 2007 respectively from Mehran UET Jamshoro, Pakistan.

Presently working as Assistant Professor in the Department of Electronic Engineering, Islamia university of Bahawalpur, Pakistan. He has more than 25 international conferences and journal publications. He is a member of PEC, IEEE, IEP, IJPE, UACEE, IACSIT, ICCTD, IACSIT, IAENG, etc. and reviewer of several journals and conferences.

He is a member of PEC, IEEE, IEP, IJPE, UACEE, IACSIT, ICCTD, IACSIT, IAENG, etc. and reviewer of several journals and conferences.

Figures



Figure 1

Some randomly selected leaf images from PlantVillage Dataset

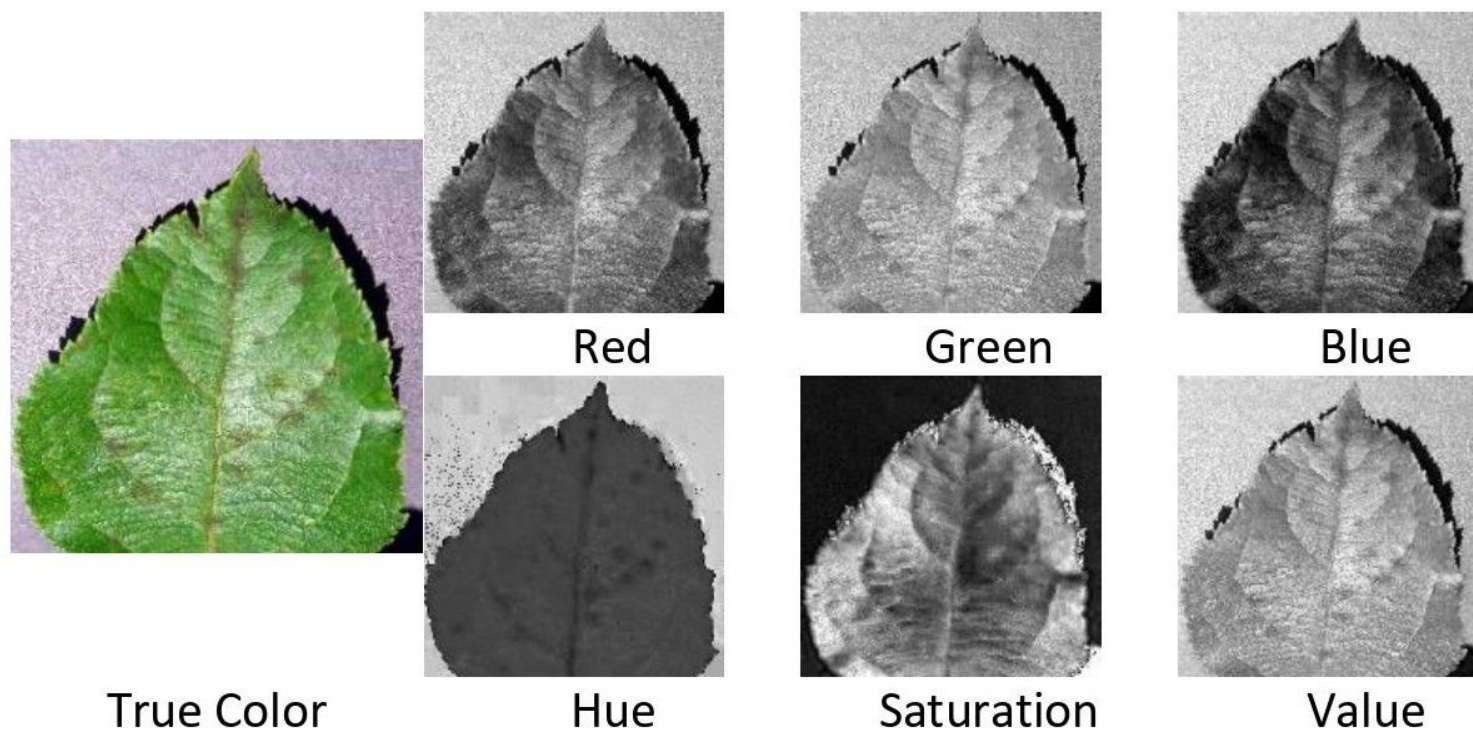


Figure 2

True Color image (Left) red, green, blue as well as hue, saturation and value channels of the image

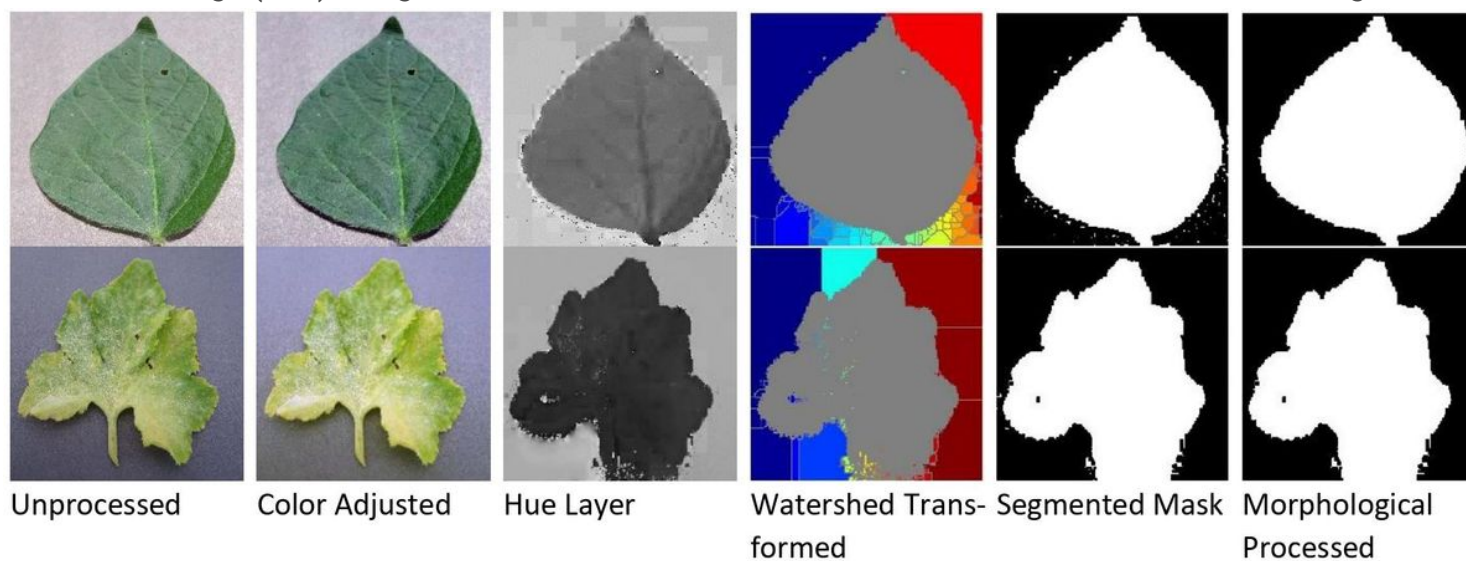


Figure 3

Steps of Preprocessing and background removal on two sample images



Figure 4

Sample Images after background removal

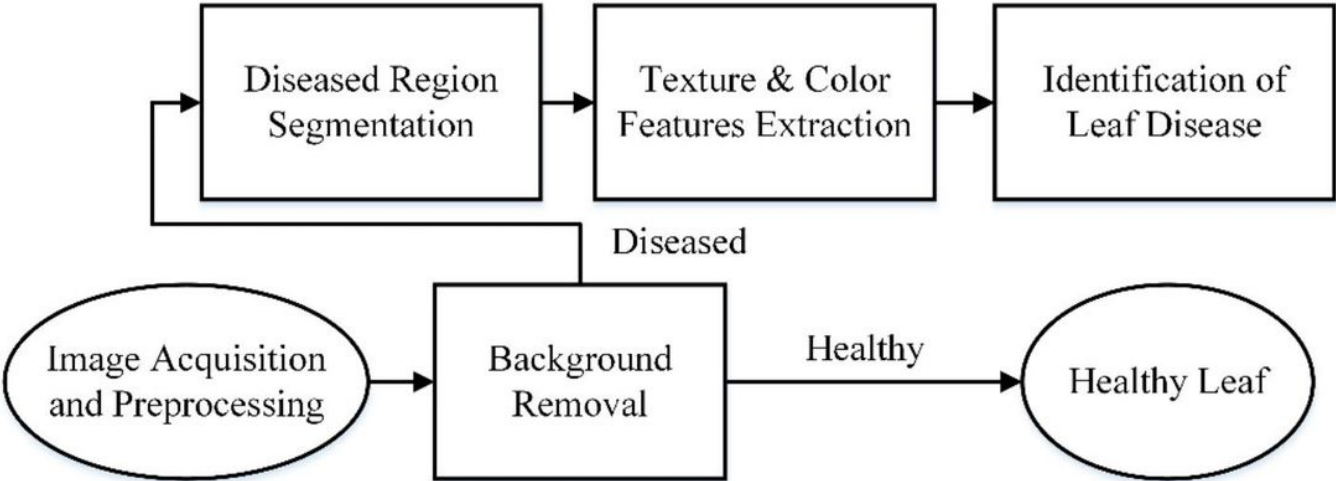


Figure 5

Flow chart of leaf disease identification



Figure 6

(Row-1) Diseased Leaf Images of Corn, Grape and Potato, (Row-2) Segmented diseased region of leaves

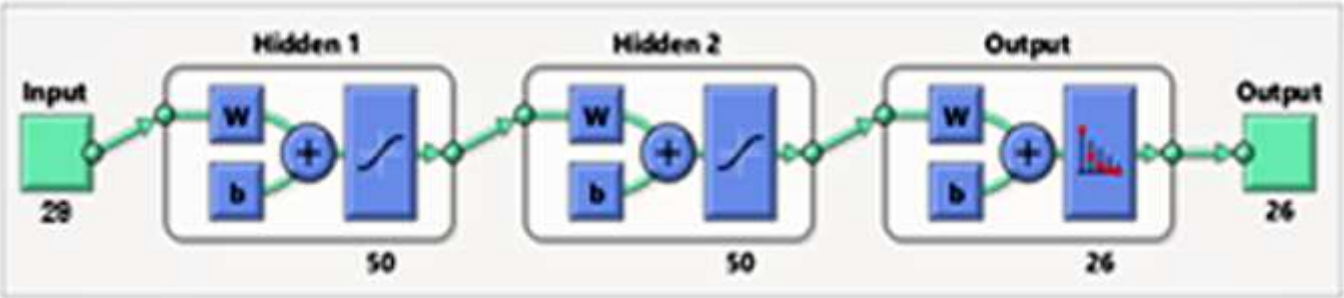


Figure 7

Artificial neural network architecture with two hidden layers