A Cloud Edge based Intelligent System for Detection of Grape Diseases

Murtaza Ali Khan
Sohar University
Muhammad Zakarya
Abdul Wali Khan University
Muhammad Haleem (m.haleem@kardan.edu.af)
Kardan University

Article

Keywords:

Posted Date: June 2nd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2992166/v1

Additional Declarations: No competing interests reported.

A Cloud Edge based Intelligent System for Detection of Grape Diseases

Murtaza Ali Khan[1], **Muhammad Zakarya**[2], **and Muhammad Haleem**[3,*]

[1]Faculty of Computing and Information Technology, Sohar University, Oman
[2]Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan
[3]Department of Computer Science, Kardan University, Kabul, Afghanistan
[*]m.haleem@kardan.edu.af

ABSTRACT

SegmentatFinding plant diseases early on is essential for reducing damage while improving the quality of the yield. This paper describes an intelligent cloud-edge-based system for identifying and categorizing diseases in grape plants. The system is based on the support vector machine (SVM), a supervised machine learning technique to classify data. The traits of healthy and diseased plants are identified using digital photographs of grape plants. To determine color and texture information, we retrieved global features from the grape images. Patterns or structures (such as corners or edges) are discovered using the speeded-up robust features (SURF) method. The K-means clustering approach is used to quantify feature space, which lowers the number of feature descriptors. The training set for the SVM classifier is made up of feature descriptors. The system classified the unlabeled grape images using the trained SVM classifier during testing. 1600 RGB pictures from four classes: Black-rot, Black-measles, Leaf-blight, and Healthy-leaf make up the original data set. To evaluate the system and provide accuracy and confusion matrices, simulations are run in four different color spaces (grayscale, RGB, YCbCr, and L*a*b*). In the L*a*b* color space, the system attained a maximum average accuracy of up to 90.63% at a ratio of 70:30 training to testing data.

1 Introduction

Early detection of plant disease is a crucial factor to make harvest healthy. The most common approach for identifying plant disease is an examination by a plant pathologist. But this approach can be time-consuming or difficult due to the unavailability of experts at the place of cultivation. With the advancement of machine learning and pattern recognition techniques, artificial intelligence systems can assist in detecting plant diseases and preventing or reducing agriculture and economic loss in the foreseeable future.

Vitaceae (Grape/Grapevine family) is cultivated worldwide, and the grape itself is used for a range of products. The grape plantation is one of the significant horticultural industries of the world, with a cultivated area exceeding 7.5 million ha ((International Organization of Vine and Wine, (1)). The grape is used in various products, ranging from fresh fruit, preserves, juice, wine, and raisins. Figure 1 shows a grapevine plant. Grapevines have evolved in various distinct world places, leading to a great many new species forming. The origin of cultivation of the V. vinifera grape, now planted throughout the world, is probably in southern Caucasia (2; 3). Almost every climate can support the grape plant. Although certain varieties do better in warm climates and some do better in cold climates. The grapevine is a climbing vine. It has therefore developed to utilize the trunks that are already present in woods and bushlands and does not need the solid support of a tree trunk. (3). About 75–85% of the weight of ripe grapes is made up of water; the remaining 15–25% is made up of sugar, which is a higher proportion than in many other fresh fruits. The organic acids tartaric, malic, and citric make up 0.5–1.0% of the fruit, and pectin proportion is about 0.25% (3). This work presents a system that employs image processing and machine learning to identify and classify healthy and disease-affected leaves in the grape plant. We used digital images of grape leaves and extracted their patterns (features). The underlying assumption is that a specific type of plant disease creates certain patterns on the leaves of a plant. The labeled images of known grape diseases and feature descriptors train a classifier using a supervised machine learning algorithm. After training, the system can test an unlabeled image of a grape leaf and classify it.

The suggested system has three crucial phases. Global and local feature descriptors are derived from the grape images in the first stage. The K-means clustering approach is then used in the next phase to decrease a

Figure 1. A grape plant. Image is free to use for a non-commercial purpose under Unsplash license (https://unsplash.com/license)

large number of feature descriptors. The classier (SVM) is then trained using a label image. The pre-trained classifier assigns a classification to an unlabeled test image.

The following is a summary of this work's main accomplishments:

1. Efficient algorithms and architecture for an automated system to identify and categorize diseases in grapes.
2. Using cloud-edge-based architecture to enhance the system efficiency and availability.
3. The system was compared in the grayscale, RGB, YCbCr, and L*a*b* color spaces with the goal of determining the best color space and parameters for spotting and categorizing grape diseases.
4. Using a more extensive data set of 1600 images compared to existing methods and establishing a statistical benchmark for future research.

This paper is structured as follows. Section 2 briefly describes the three pathogens (diseases), i.e., Black-measles, Black-rot, and Leaf-blight of the grape plant. An overview of three color spaces is included in section 3. We outline the architecture and algorithms of the suggested system in section 6. Section 8 gives details of the simulation and results. A discussion about the system and results are the topic of section 9. Limitations of the proposed system and possible future work are highlighted in section 10. Section 11 reports the final concluding remarks.

2 Pathogens of grape plant

The grape plant has several types of pathogens caused by fungus, bacteria, or viruses. Identification and treatment of grape diseases are essential because of plant health reasons and the economy. It's important to limit grape exposure to harmful effects. Depending on the circumstances, a substantial proportion of vineyard costs can be devoted to managing diseases. The trunk, arms, cordons, canes, shoots, leaves, rachis, and berries of grapevines can all sustain damage. The three different disease types—Black measles, Black rot, and Leaf blight—are discussed in this paper. Figure 2 shows disease-affected and healthy leaves of the grape plant in RGB, YCbCr, and L*a*b* color spaces. The following subsections briefly describe the three pathogens.

2.1 Black-measles or Esca (Phaeomoniella aleophilum)

Symptoms of Black-measles appear on leaves, trunks, canes, and grapes. Interveinal striping resembling tiger stripes appears on the leaves. Severe infestation of Black-measles can ruin the vineyard completely. Improper drainage, poorly prepared soil, and plant holes also contribute to the development of Black-measles. Soaking infected dormant cuttings or grapes in fungicides before grafting or planting reduces the severity of the disease.

2.2 Black-rot (Guignardia bidwellii)

Black-rot is a dangerous disease to vineyard farmers, and sometimes losses can reach 80 percent. Symptoms appear on leaves one to two weeks after the infection as circular lesions of yellow or cream color. Then the spots turn to darken to brown-black or reddish-brown. Black-rot fungus disease favors warm, humid, and rainy weather. This pathogen condition is controlled by removing the infected area or applying mancozeb or azoxystrobin fungicides.

2.3 Leaf-blight (Pseudocercospora vitis)

Leaf-blight symptoms appear late in the season. Irregularly shaped 2 - 25 mm diameter lesions emerge on the grape leaves. Initially, lesions are brown or dull red; later, they turn black. If the disease is severe, then these lesions may coalesce. Some grape varieties, such as Cynthiana and Cabernet Sauvignon are more vulnerable to the Leaf-blight.

3 Color spaces

A color space defines how colors are represented and arranged. In this study, we analyzed the classification of grape images in grayscale, RGB, YCbCr, and L*a*b* color spaces. The following subsections briefly describe the three color spaces.

3.1 Grayscale space

Strictly speaking, grayscale is not a color space because there is no color in it!. Instead, grayscale specify intensity information – for digital images, how bright is a particular pixel. The higher the value, the greater the intensity. It is common to store each pixel using 8 bits to describe 256 levels of grey, i.e., a value from 0 to 255.

3.2 RGB color space

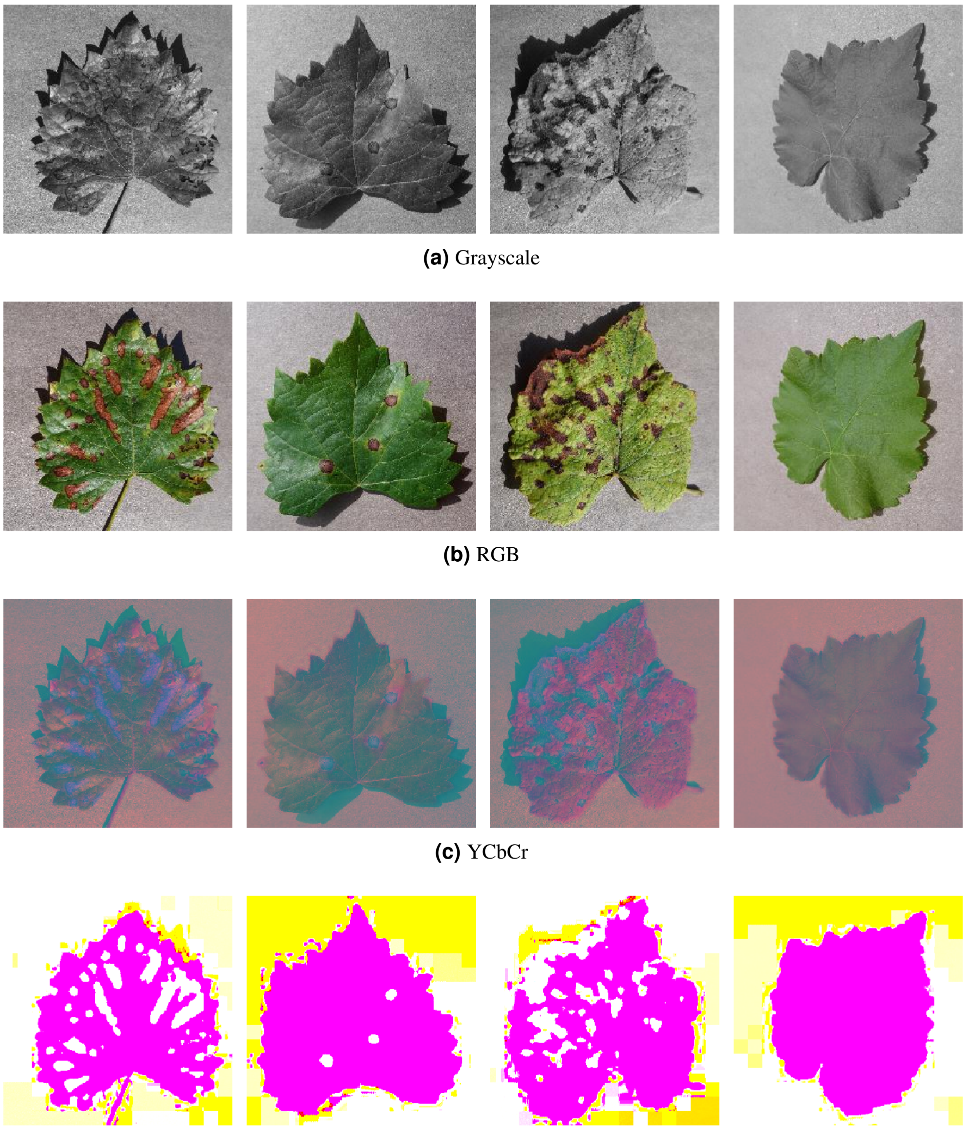
It is an additive color space featuring red, green, and blue as its three primary colors. A single color is produced by combining the intensities of three different components. Red, green, and blue axes make up a three-dimensional cube that represents the RGB color space. A point in the cube can be any color. RGB images typically require 24-bits per pixel (8-bits for each primary color, an integer value between 0 and 255). The RGB color model is simple, and most computer graphics images use it.

3.3 YCbCr color space

In the YCbCr color space, Y is the luma component, i.e., brightness, while Cb and Cr are the blue-difference and red-difference chroma components. Luma represents the image without any color, while the chroma channels express the color information. YCbCr color space assists in compression in digital video encoding because the human eye is more sensitive to luminance; the Y value is stored with greater accuracy than Cb and Cr.

3.4 CIE L*a*b* color space

The hue L* in the L*a*b* color space stands for lightness. The other two components, a* and b*, are coordinates for chromaticity. The a* and b* components are not rigorously restricted, however the value of the L component ranges from 0 to 100. The a* and b* can have values between 127 and 128. Red and green blends cannot be encoded in the L*a*b* color space. Additionally, it is impossible to concurrently encode yellow and blue. While this would appear to be a drawback, it's noteworthy to note that the human visual system has some of these qualities.



(a) Grayscale

(b) RGB

(c) YCbCr

(d) L*a*b*

Figure 2. Sample images of disease-affected and healthy grape leave in various color spaces. In each row, left to right images are of classes: Black-measles, Black-rot, Leaf-blight, and Healthy.

4 Relevant Work

Artificial intelligence (AI) applications and usage are expanding in a variety of sectors, including NLP (4; 5; 6), analysis of medical images (7; 8; 9), and identification of fruits (10). The authors of (11) offered a broad overview of methods for employing image processing to identify and categorize plant leaf diseases.

The author (10) introduces a method based on image processing for locating cucumbers in a greenhouse. In this method, the original RGB image is converted to grayscale and HSI color space. The grayscale image is used for texture analysis. The HSI space image is used to analyze the color and then remove soil and branches from the background. The method uses the scale-invariant feature transform for feature detection and SVM for classification. A technique to assess the degree of Phytophthora root rot disease damage in the avocado plant is described by (12). This method captures the RGB images of the plants that may have been damaged due to Phytophthora root rot disease. Then multi-spectral satellite images are used to measure Phytophthora acuteness that is grouped into three levels.

The authors (13) presented a color sensing and image processing-based technique to gauge the severity of foliar disease on soybean plants. There approach collects YCbCr channels from the input RGB image and then applies opening and closing bi-level morphological operations on the smooth area of the diseased region.

A mobile client-server architecture for detecting of leaf diseases in plants is described by (14). The gray-level co-occurrence matrix and the Gabor wavelet transform (GWT) are used in the approach. The smartphone acts as a mobile client in this technique. It takes a picture of a leaf and does preliminary processing to separate the diseased afflicted areas. The Global System for Mobile Communications (GSM) is then used to send images of the sick areas impacted by the illness to the distant server. The server performs classification, identifies the disease, and extracts the segmented image features using the K-means algorithm. Finally, the mobile client receives the results.

A few classic algorithms to extract patterns/features from images are the Difference of Gaussians (DoG) (15), Laplacian-of-Gaussians (LoG) (16), Harris corner detector (17), etc. The literature has a variety of descriptor-finding techniques, e.g., speeded-up robust features (SURF) (18; 19), scale-invariant feature transform (SIFT) (20; 21), and Gradient Location and Orientation Histogram (GLOH) (22), etc. We used the average of color channels in image blocks to extract global and SURF to detect local feature descriptors from the grape images.

We choose the support vector machine (SVM) for the classification of grape diseases. Vladimir Vapnik and his colleagues developed the supervised machine learning method known as SVM when they were employed by AT&T laboratories. (23), (24), (25) et al. Some of its advantages are minimum generalization error, low processing expense, and straightforward results interpretation. SVM is very popular among AI and data scientists to solve various problems, from medical diagnosis to image recognition and textual classification. The work of (26) proposed an iterative method of keypoint selection in the bag-of-words model. The method can use for image classification using an SVM classifier. (27) presents a semi-supervised support vector machine classifier (S^3VM) built on active learning and context data. In this approach, a semi-supervised learning method initially uses active learning to select unlabeled samples as semi-label samples. Then further expansion and re-labeling of the chosen samples are performed using context information. The most important works that are connected to our study are included in Table 1 along with the benefits and drawbacks of each strategy.

Table 1. Table of related works with the pros and cons of each approach.

References	Pros	Cons
(10)	The study identifies cucumbers fruits in the greenhouse	The method assumes there are no defects or diseases in the cucumbers, and manual intervention is needed for segmentation
(28)	This method detects diseases of potatoes using image segmentation and an SVM classifier	The method can classify only two types of potato disease and uses a dataset of only 300 images
(12)	This work quantifies the severity of Phytophthora root rot disease in avocado trees using satellite images	The technique is complex and depends on spectral satellite images that may not be available to most farmers
(14)	The methods detect leaf disease using a smartphone camera via a mobile client-server architecture	a farmer with a low-quality camera or lack of taking good pictures skills may not get the correct results
(13)	The methods detect soybean plant foliar disease via image processing techniques	Removal of background may produce incorrect results, and the system's performance could be improved by using advanced background separation methods

5 Cloud and Edge based System Architecture

This section elaborates on the architecture of the system to detect and classify diseases of the grape plant. To distribute and execute the suggested models independently over an edge-cloud architecture, we segment the models into several modules, as shown in Figure 3. While the testing modules operate on the edge,

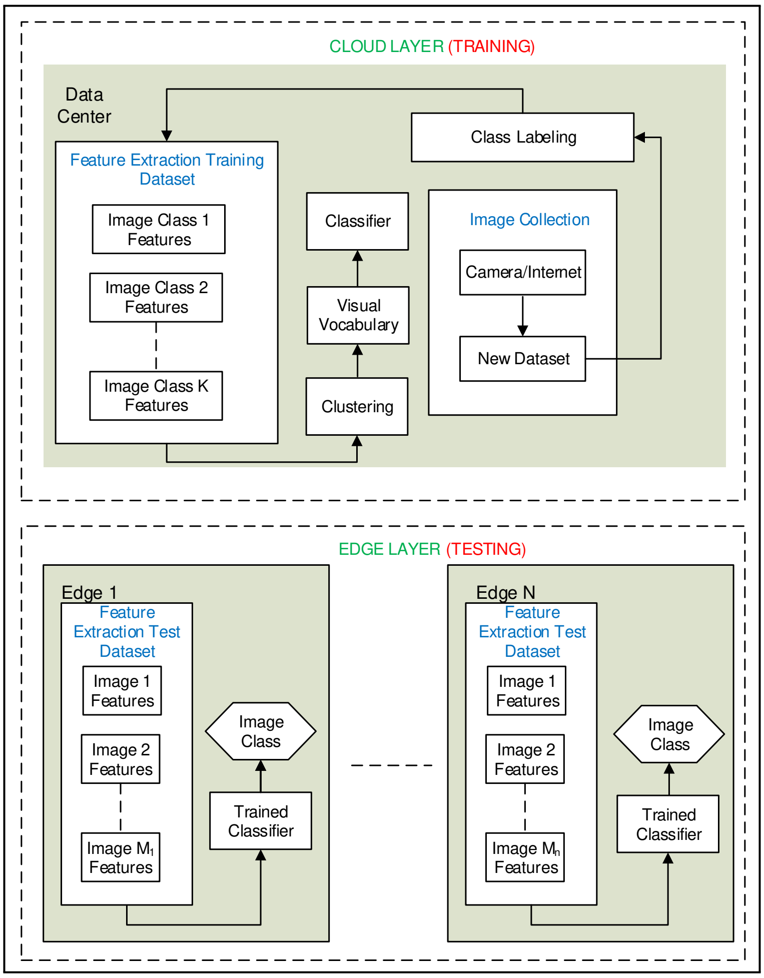


Figure 3. System architecture.

the training and feature extraction modules operate in a remote cloud. It is possible to increase the model efficiency and latencies by training an SVM model and extracting features on the cloud, using the trained SVM classifier and validating on the edge, and finally. The main components of our suggested model are listed below.

5.1 Data Collection

In this module, we gather different datasets from edge devices or other IoT instruments and send them to the cloud for additional processing, such as training and feature extraction. To create training, validation, and/or test samples, divide the dataset.

5.2 Feature Extraction

Using global and local feature extraction methods, this component is in charge of extracting features from the training and testing dataset. The cloud computer is used for this stage because it calls for a lot of computational power.

5.3 Model Training

In this component, we train the labeled dataset and built the visual vocabulary of features. The cloud computer is used for this stage because it calls for a lot of computational power. The SVM model is disseminated to the edge devices for inference after it has been learned.

5.4 Classifier

Using the trained SVM model and the features derived from the validation or test set, we perform classification on the edge devices in this module. Because it needs little communication overhead and delay, this stage is carried out on edge devices. This module may also be used to assess the SVM model's precision using the test or validation set. Depending on the available processing resources and communication cost, this phase may be carried out on edge devices or the cloud server.

5.5 Testing

The testing module is implemented at the edges. In this module, the system determined the classes of unlabeled images using the pre-trained SVM classifier. A separate copy of classifier is installed at each edge.

We can take advantage of the cloud server's computational capabilities to carry out highly demanding jobs by training the SVM model and carrying out feature extraction in the cloud. We decreased the transmission overhead and delay associated with moving data to and from the cloud server by conducting classification on the peripheral devices, i.e., edges. Additionally, we can allow edge devices to conduct real-time Classifier without depending on the cloud server by spreading the learned SVM model to the edge devices.

6 System Functionality and Algorithms

This section provides details on how the various modules of the system work in an integrated manner. During training, a set of each class of grape images is fed to the system. The feature extraction block extracts the localized and global features of the image sets. The visual vocabulary is formed via the K-means clustering method used to train the SVM classifier. Identifying an unlabeled grape image class involves detecting its features and invoking the pre-trained SVM classifier. Algorithm 1 and Algorithm 2 describe the testing and training procedures of the system. The following sub-sections give the details of the extraction of feature descriptors, clustering, and classification.

Algorithm 1 Training algorithm

Require:

C_L: Labeled colored image sets

G_L: Labeled grayscale image sets

Ensure:

$ECOC$: Error-correcting output codes

FD: Feature-descriptors of labeled image sets

VV: Visual-vocabulary of labeled image sets

for Each labeled image-set **do**

$$FD \leftarrow \begin{cases} MeanOfBlocks(C_L), \\ SURF(G_L), \end{cases} .$$

$VV \leftarrow K$-means(FD)

$ECOC \leftarrow$ SVM-Classifier(VV, C_L, G_L)

end for

Algorithm 2 Testing algorithm

Require:

C_U: Unlabeled colored image sets

G_U: Unlabeled grayscale image sets

$ECOC$: Error-correcting output codes

Ensure:

LC: Leaf-classes of unlabeled image

FD: Feature-descriptors of an unlabeled image

VV: Visual-vocabulary of an unlabeled image

for Each image in the unlabeled image-sets **do**

$$FD \leftarrow \begin{cases} MeanOfBlocks(C_{U_k}), & \text{kth color image.} \\ SURF(G_{U_k}), & \text{kth grayscale image.} \end{cases}$$

$VV \leftarrow K$-means(FD)

$LC \leftarrow$ SVM-Classifier($ECOC$, VV, C_{U_k}, G_{U_k})

end for

6.1 Extraction of feature descriptors

In an image, a feature is the point or region of interest. Feature points are immune to various transformations such as rotation, scaling, and changes in light intensity. The points are chosen from high contrast regions. A *feature descriptor* is a vector of values that describes the image patch around the point of interest. In a classification task, feature descriptors of a test image compared with features of trained images and closest correspondence is selected as the best match. Features can be global or local. Global features express the image as an entire. In contrast, the local features focus on detecting critical points within the parts of the image. Statistical techniques, e.g., mean or quantization, can find global features, while SURF and SIFT are two widely used algorithms to extract the local features.

We extracted the global feature descriptor of grape images in three color spaces, i.e., RGB, YCbCr, and CIE L*a*b*. The mean of a block (32×32 pixels) for each channel is used as a feature descriptor for that block. Though it seems a simple method, yet proved to be very fast and gives excellent results.

Local features better characterize grayscale images that have only one channel that holds the luminance value. We employed the SURF algorithm to detect the local feature descriptor of grape images. For the detection stage of keypoint points in grape images, the SURF computation is based on simple 2D box filters to approximate the second-order Gaussian deriva- tives efficiently. Furthermore, it uses a scale-invariant blob detector based on the determinant of the Hessian matrix for both scale selection and locations. After detecting feature descriptors from a grape image at a pixel location $p(x,y)$, scale s, and orientation θ, the image structure in the vicinity of p is encoded.

6.2 Clustering of feature descriptors

The SURF algorithm finds thousands of feature descriptors of grape images. In the next step, the system reduces these descriptors into a mutually exclusive small set using the K-means clustering algorithm with $K = 500$. The center of each cluster represents a visual word. The centers of all the clusters of all the images are collectively called the visual vocabulary. The K–means clustering algorithm can be summaries as follows:

1. Randomly assign K points as starting centroids of the feature descriptors of grape images
2. Assign each feature descriptor to a cluster that minimizes the distance between the descriptor and the centroid
3. Compute the centroid of each cluster again by taking the average of descriptors in that cluster
4. Return to step 2 until any descriptor does not change its centroid

6.3 Classification of grape images

During training, the SVM classifier learns the features of each grape class and builds a visual vocabulary. During testing, the classifier matches the features of the unlabeled test grape image with the visual vocabulary of the classifier then predicts the class of the test grape image based on the optimal match.

The two classes SVM setup an optimal hyperplane in a high dimensional space that separates sample points into two classes labeled as -1 and 1. The best-chosen hyperplane maximizes the separation between two classes. The separation is defined as the perpendicular distance between the decision boundary and the closest of the sample points. Mathematically, hyperplane H can be defined as follows:

$$H : w.x - b = 0, \tag{1}$$

where $b \in \mathbb{R}$ is called the bias, and $w \in \mathbb{R}^n$ is referred to as the weight vector. Let sample data is a set of points x_i with their classes $y_i \in [1,-1]$. Following conditions are satisfied for every point $i = 1,2,\ldots,m$,:

$$w.x_i - b > 0, y_i = 1, \tag{2}$$

$$w.x_i - b \leq 0, y_i = -1. \tag{3}$$

The optimal solution needs to find the value of (b,w). The above equations can be solved using standard techniques such as (29), (30), (31) et al. To solve the multiclass problem, as in our case, four grape classes. Multiple binary SVM is used such that each SVM separates one from the rest of the classes. A common approach to building m-SVM is to reduce the single multiclass problem into multiple binary classification problems. Our system built an m-SVM model by training it for grape images of four classes.

7 Terminologies

Let X be one of the image classes (Black-measles, Black-rot, Leaf-blight, or Healthy-grapes). Then following terminologies are used to measure the accuracy of our system.

- True-positive (TP) is the number of correctly classified grape images as of class X.
- True-negative (TN) is the number of correctly classified grape images as NOT of class X.
- False-positive (FP) is the number of incorrectly classified grape images as of class X.
- False-negative (FN) is the number of incorrectly classified grape images as NOT of class X.
- Accuracy is the fraction between correctly classified grape images to the total number of classified grapes images (correctly and incorrectly), mathematically, $Accuracy = (TP+TN)/(TP+TN+FP+FN)$.

8 Results

A public dataset of 1600 images (32) of grape leaves is used in the simulation. The PlantVillage dataset (32) (https://www.kaggle.com/emmarex/plantdisease) consists of diseased plant leaf images of various plants and corresponding labels. Although leaf images of various plants such as pepper, potato, tomato, etc., are available in this dataset, we have downloaded grape images. Furthermore, we uploaded all chosen images to site (32) for easy accessibility.

There are four classes of images: (1) Black-measles, (2) Black-rot, (3) Leaf-blight, and (4) Healthy-leaves. The Healthy-leaves class does not have any disease. Each class has 400 images. We intentionally chose an equal number of samples in each set, such that the classifier remains unbiased towards any category of grape leaves. The original dataset is in RGB color space. We converted the images (on the fly) into grayscale, YCbCr, and L*a*b* color spaces. The system is implemented on MATLAB R2019a programming language on Windows 10 Home 64-bit operating system. The hardware consists of an Intel Core i7 7700HQ 2.8GHz processor supported by 16GB of RAM.

We randomly divide the grape data into training and testing parts in various ratios. For example, if $P\%$ of the grape images are taken as the testing set, then the remaining $(100-P)\%$ of grape images are part of the training set. We have a total of 1600 grape images. For example, at $P = 30\%$, there are 480 images in the training set and 1120 images in the testing set. Out of 480 images of the training set, each class of grape leaves has 120 images. There are 280 images in each of the grape classes out of 1120 images of the testing set. In each round of the simulation, the value of the training: testing ratio changes. The first row of Table 2 lists the various values of training data used in the simulation. In Table 2, the second and third rows refer to the count of images in the training and testing sets. The fourth row of Table 2 lists the number of features obtained from the training set of grape leaves. We discard the 20% week features and retain the 80% robust features (see the fifth row of Table 2). The 80% robust features are passed to the K-means clustering algorithm that creates 500 mutually exclusive clusters from these features. The cluster's center represents a visual word, and all visual words add up to the bag of visual words. Next, the bag of visual words of grape leaves is fed to the multiclass SVM classifier to train it. After completion of SVM training, its accuracy is evaluated using the testing set of grape images. In the testing set, the classes of grape leaves are not known. For each grape image in the testing set, its feature descriptors are obtained and grouped into clusters to construct the visual vocabulary that enables the SVM to forecast the class of each image.

Table 2. Details of features extraction from 400×4 images for $K = 500$ (for K-means clustering).

Training data%	20	30	40	50	60	70	80
No. of training images	320	480	640	800	960	1120	1280
No. of testing images	1280	1120	960	800	640	480	320
No. of features	81920	122880	163840	204800	245760	286720	327680
No. 80% strongest features	65536	98304	131072	163840	196608	229376	262144

Table 3 provides details of comparative classification accuracy in grayscale and three color spaces for four grape leaf classes at various training data values. Figure 4 is the visual representation of the results of Table 3. The mean accuracy in a given color space is the average accuracy of all the grape classes. Table 4

Table 3. Comparative classification accuracy in grayscale and three color spaces at various values of training data. Four types of grape leaves are: *BM* $\rightarrow$ Black-measles, *BR* $\rightarrow$ Black-rot, *LB* $\rightarrow$ Leaf-blight, and *HL* $\rightarrow$ Healthy-leaves.

% Training data	**% Accuracy in Grayscale**				**% Accuracy in RGB**			
	BM	*BR*	*LB*	*HL*	*BM*	*BR*	*LB*	*HL*
20	71.25	26.87	86.88	92.50	87.50	58.44	88.75	95.94
30	78.21	25.71	86.07	93.93	83.93	67.86	92.14	95.36
40	75.42	33.33	86.67	92.50	85.42	70.42	87.50	97.92
50	73.50	36.00	85.00	95.00	82.50	70.00	92.50	99.00
60	82.50	30.63	85.62	91.25	88.75	75.62	88.12	98.75
70	75.00	45.83	89.17	94.17	87.50	80.00	90.83	95.83
80	76.25	38.75	93.75	91.25	90.00	67.50	90.00	98.75
% Training data	**% Accuracy in YCbCr**				**% Accuracy in L*a*b***			
	BM	*BR*	*LB*	*HL*	*BM*	*BR*	*LB*	*HL*
20	83.75	46.88	87.19	96.88	83.75	71.56	89.69	97.50
30	82.86	53.21	85.71	97.50	86.43	72.14	91.07	96.79
40	82.92	53.75	87.92	97.50	87.92	71.67	90.42	97.92
50	78.50	56.00	86.50	96.50	88.50	79.00	88.50	99.00
60	84.38	57.50	86.88	98.75	87.50	75.00	90.63	98.75
70	86.67	62.50	90.83	96.67	90.83	80.83	91.67	99.17
80	82.50	57.50	82.50	100.00	81.25	81.25	92.50	98.75

provide details of the mean accuracy in grayscale and three color spaces at various values of training data. Figure 5 shows mean accuracy as line graphs in grayscale and three color spaces at various values of training data.

Figure 6 shows confusion matrices (CMs) in three color spaces at 50% training data. Two hundred images are used in training, and the remaining 200 are part of the testing set. Each CM has four rows and four columns. A value in a cell of CM shows the number of grape images classified. Rows correspond to the predicted class determined by the system, while the columns correspond to the actual class as given in the known dataset. The diagonal cells of the CM represent the number of correctly classified plant images. The off-diagonal cells show the misclassified plant class. Empty cells in Figure 6 have the value 0.

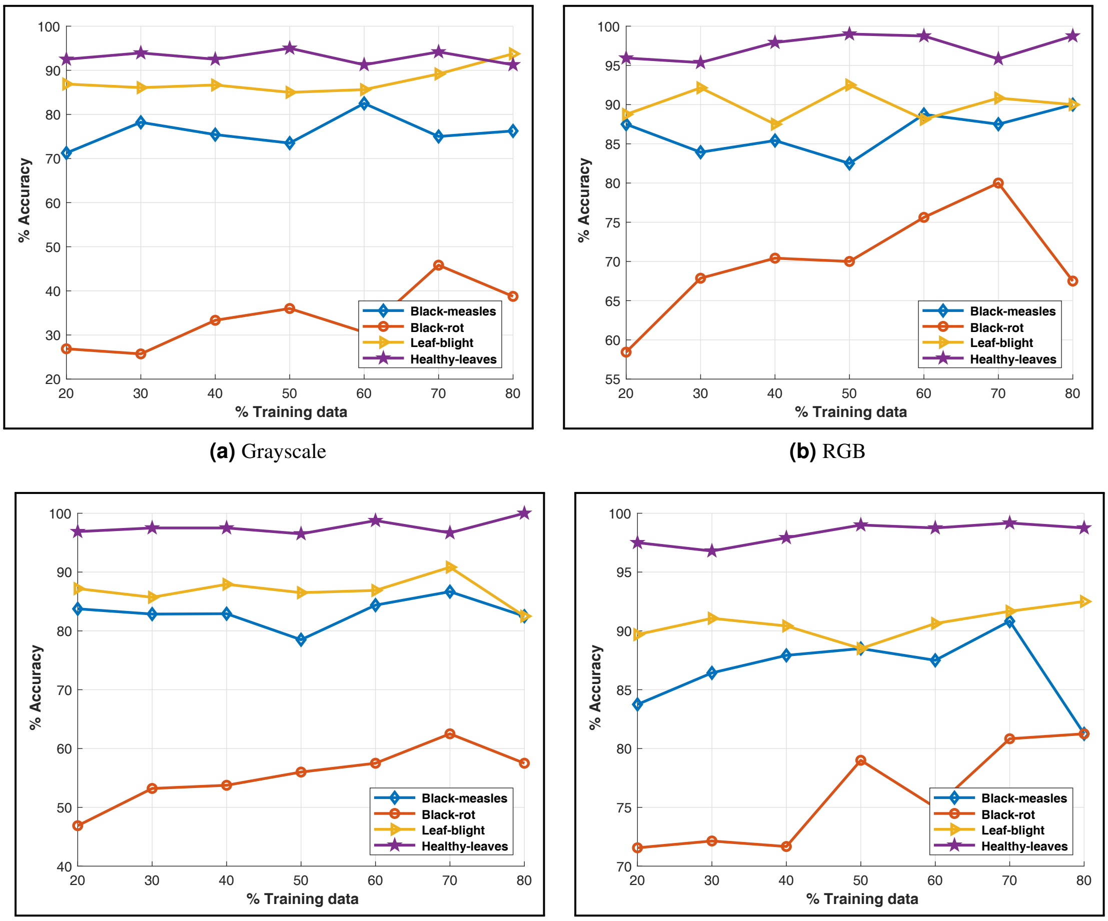
Table 4. Mean accuracy of classification for all the classes of grape leaves in grayscale and three color spaces at various values of training data.

% Training data	% Mean Accuracy in Grayscale	RGB	YCbCr	L*a*b*
20	69.38	82.66	78.67	85.63
30	70.98	84.82	79.82	86.61
40	71.98	85.31	80.52	86.98
50	72.38	86.00	79.38	88.75
60	72.50	87.81	81.88	87.97
70	76.04	88.54	84.17	90.63
80	75.00	86.56	80.63	88.44

9 Discussion

9.1 Analysis of accuracy

Table 3 and Figure 4 suggest that 70% of the training data yields the maximum classification accuracy. However, 80% of training data causes over-fitting, i.e., excessive training data decrease the accuracy of three types of grape leaves in YCbCr color spaces and two classes in grayscale, RGB, and L*a*b* color spaces. We can infer from Figure 3 that the classification accuracy of Healthy-leaves is highest in all three color spaces because healthy- leaves have the most uniform or consistent pattern due to the absence of any disease.



(a) Grayscale **(b)** RGB

(c) YCbCr **(d)** L*a*b*

Figure 4. Classification accuracy in grayscale and three color spaces at various values of training data. The scaling of X-axes is the same, while the scaling of Y-axes is adjusted.

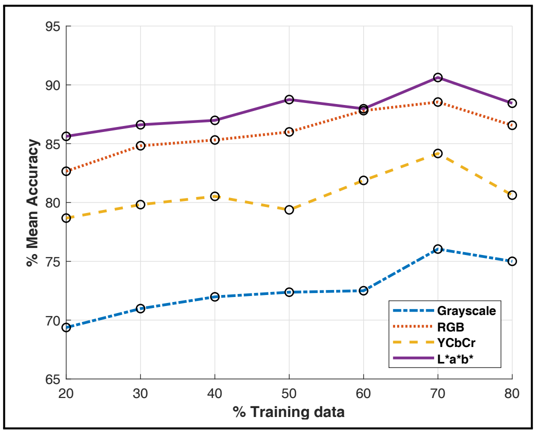
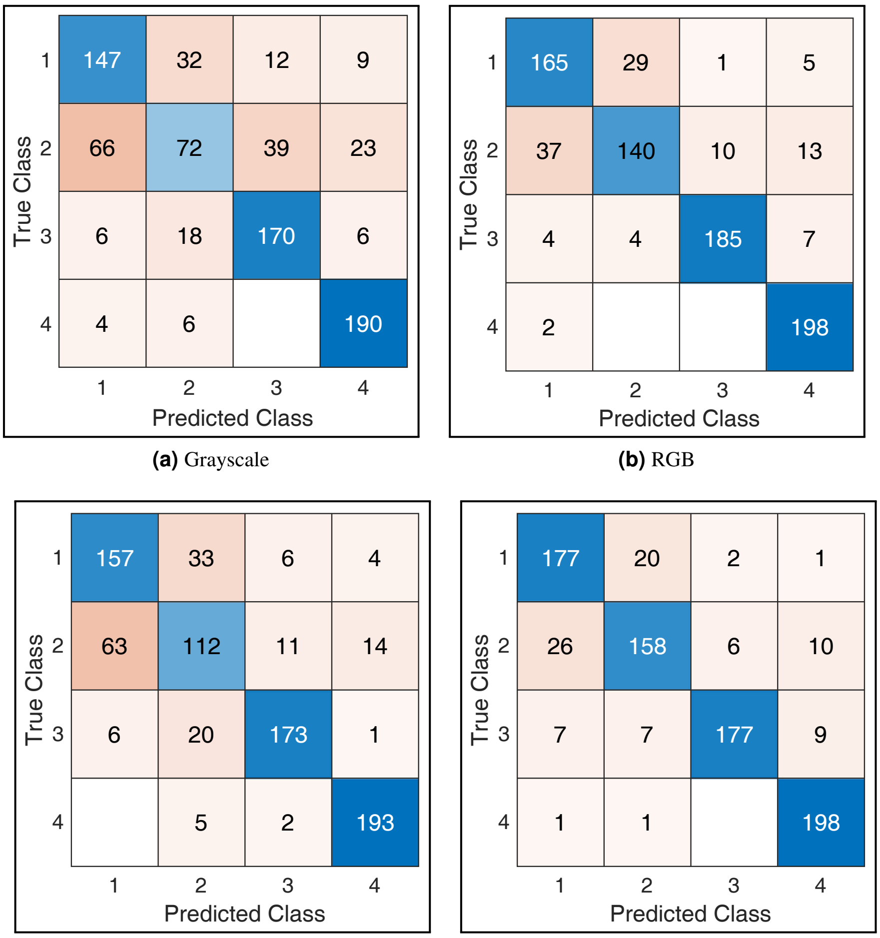


Figure 5. Mean accuracy as line graphs in grayscale and three color spaces at various values of training data.

In contrast, a grape disease induces patches of various sizes and colors in the grape leaves. The larger the number of colors in an image, the more scattered its data in the color space, and consequently, it becomes challenging for the classifiers to identify the pattern. Therefore, fewer variations of Healthy-leaves' shades assist in their clustering, and thus, their classification accuracy is better than disease-affected grape leaves.



(a) Grayscale **(b)** RGB **(c)** YCbCr **(d)** L*a*b*

Figure 6. Confusion matrices in grayscale and three color spaces at 50:50 Training:Testing ratio. Class labels: 1→Black-measles, 2 → Black-rot, 3 → Leaf-blight, 4 → Healthy-leaves.

Further, the perceptual linearity of L*a*b* color space gives more support to Healthy-leaves in L*a*b* color space than grayscale, RGB, and YCbCr color spaces. Therefore, the accuracy of Healthy-leaves is highest in the L*a*b* color space.

9.2 Analysis of confusion matrices

The CMs of Figure 6 belong to 50:50 training: testing data, i.e., out of 400 images of each grape class, 200 are randomly chosen for training and the remaining 200 for testing. Statistics shown in the figure are related to testing data. The second CM in Figure 6 belongs to the RGB color space. For example, in row 2 of this CM, out of 200 $(37+140+10+13)$ grape images, the model classified 37 as Black-measles, 140 as Black-rot, 10 as Leaf-blight, and 13 is Healthy-leaves. So, out of 200 grape images, 140 are correctly classified as Black-rot. Note that the correctly classified number, i.e., 140 is written in the diagonal cell (2, 2), as the Black-rot class has label 2. Now let's take an example of L*a*b color space, i.e., rightmost CM of Figure 6. In row 3 of the CM, out of 200 $(7+7+177+9)$ grape images, the system classifies 7 as Black-measles, 7 as Black-rot, 177 as Leaf-blight, and 9 as Healthy-leaves. The correctly identified Leaf-blight images are 177, written in the diagonal cell (3, 3). Consequently, the accuracy in identifying Leaf-blight grape images in L*a*b color space at 50% training data is $(177/200)\times 100 = 88.5\%$. Confusion matrices also indicate that the false negative (FN) rate of the Black-rot class is highest. A substantial number of Black-rot images are classified as Black-measles, i.e., cell (2, 1) of CMs in all the three color spaces.

9.3 Segmentation

We experimented with image segmentation to isolate disease-affected parts of the grape leaf with the healthy area before applying the feature extraction step. Segmentation allows the classification algorithm to use only features of the disease-affected part of the grape leaf. However, we found that segmentation decreases the accuracy of the system. Therefore, we omitted the segmentation step from our system (Fig. 2). Several times the segmentation algorithm yields images where disease affected area is not entirely isolated. Consequently, segmentation is not fully automated, and human intervention is needed. Our aim is that the end-user does not need to know about image segmentation to fine-tune images for feature extraction or classification.

9.4 Background Subtraction

The grape leaf dataset we used has a uniform background for all the classes of images, as these photos are taken in a laboratory setting. Therefore, we did not subtract the background. However, samples are likely to have different backgrounds in more real-life situations. In such a situation, the subtraction of background will assist the classifier in distinguishing classes.

10 Limitations and Future Work

The proposed system identifies three common grape leaf diseases, i.e., Black-measles, Black-rot, and Leaf-blight. But there are other types of conditions, such as Downy-mildew. Further, we investigated leaf diseases, but some grape disease affects the berries (fruit) such as Green-mould-rot. In the future, a more comprehensive system needs to be developed to identify a range of ailments that harm both leaves and berries. We employed an SVM classifier that is fast and efficient. Implementing the system using other machine learning algorithms and classifiers, such as decision trees, random forests, and deep learning neural networks, will provide better comparative insight.

There are significant difficulties in distributing the SVM model to the edge devices, such as assuring compatibility and tailoring the model to the various hardware and software setups of the edge devices. Furthermore, it is crucial to guarantee data privacy and security while sending and storing sensitive data on edge devices and the cloud. In general, the method used to train an SVM model, carry out feature extraction in the cloud, and use the learned SVM classifier for classification and testing on the edge relies on the particular needs and restrictions of the application. In the future, we will create a reliable and effective system that makes use of the advantages of both cloud and edge devices by carefully balancing the trade-offs between accuracy, latency, privacy, and scalability.

11 Conclusion

Grape is an important crop that provides income to millions of people worldwide. Unfortunately, diseases in grape plants are caused by various organisms, including bacteria, fungi, and viruses. Therefore, identifying and treating grape diseases at the right time can substantially reduce financial loss. This work presents a cloud-edge based intelligent color space optimized system to detect and classify healthy and disease-affected grape images of three types of diseases: black-measles, black-rot, and leaf-blight. Grape images are initially fed to the system to split into training and testing sets. Local features of grayscale images are obtained via the SURF algorithm. Global feature descriptors are extracted from the color images using the mean value of 32×32 pixels block. The K-means clustering algorithm constructs the visual vocabulary by grouping descriptors into clusters. The multiclass support vector machine (SVM) classifier is trained using feature descriptors. When testing an unlabelled image, the system extracts its features, builds the visual vocabulary, and sends it to the pre-trained SVM classifier. We tested the system using 1600 images in grayscale, RGB, YCbCr, and L*a*b* color spaces. The research determined that L*a*b* is the most suitable color space for classifications. We also found that $K = 500$ is the most appropriate value for the K-means clustering algorithm. The proposed system yields the maximum average accuracy of 90.63% at 70:30 training: testing data ratio in the L*a*b* color space.

Declarations

Ethical Approval

Not applicable. The paper does not have human and/or animal studies.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions

Murtaza Ali Khan:- Research, Conceptualization, Methodology, Writing - Original Draft;
Muhammad Zakarya:- Visualization, Data Curation, Proofreading;
Muhammad Haleem:- Writing - Revised Draft, Data Curation;

Funding

No funding received

Availability of data and materials

The data set is available at the following share repository:
https://figshare.com/ndownloader/files/37001836

References

1. International organisation of vine and wine. https://www.oiv.int/.

2. Mullins, M. G., Bouquet, A. & Williams, L. E. *Biology of the grapevine* (Cambridge University Press, 1992).

3. Creasy, G. L. & Creasy, L. L. *Grapes* (CABI, 2018).

4. Akstinaite, V., Garrard, P. & Sadler-Smith, E. Identifying linguistic markers of CEO hubris: A machine learning approach. *Br. J. Manag.* DOI: 10.1111/1467-8551.12503 (2021).

5. Umer, M. *et al.* Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model. *Comput. Intell.* **37**, 409–434, DOI: 10.1111/coin.12415 (2020).

6. AlGhamdi, M. A. & Khan, M. A. Intelligent analysis of arabic tweets for detection of suspicious messages. *Arab. J. for Sci. Eng.* **45**, 6021–6032, DOI: 10.1007/s13369-020-04447-0 (2020).

7. Sha, Z., Hu, L. & Rouyendegh, B. D. Deep learning and optimization algorithms for automatic breast cancer detection. *Int. J. Imaging Syst. Technol.* **30**, 495–506, DOI: 10.1002/ima.22400 (2020).

8. Singh, A. *et al.* An intelligent hybrid approach for hepatitis disease diagnosis: Combining enhanced k-means clustering and improved ensemble learning. *Expert. Syst.* **38**, e12526 (2021).

9. Khan, M. A. An automated and fast system to identify COVID -19 from x-ray radiograph of the chest using image processing and machine learning. *Int. J. Imaging Syst. Technol.* **31**, 499–508, DOI: 10.1002/ima.22564 (2021).

10. Li, D., Zhao, H., Zhao, X., Gao, Q. & Xu, L. Cucumber detection based on texture and color in greenhouse. *Int. J. Pattern Recognit. Artif. Intell.* **31**, 1754016, DOI: 10.1142/s0218001417540167 (2017).

11. Dhingra, G., Kumar, V. & Joshi, H. D. Study of digital image processing techniques for leaf disease detection and classification. *Multimed. Tools Appl.* **77**, 19951–20000, DOI: 10.1007/s11042-017-5445-8 (2017).

12. Salgadoe, A. S. A., Robson, A. J., Lamb, D. W., Dann, E. K. & Searle, C. Quantifying the severity of phytophthora root rot disease in avocado trees using image analysis. *Remote. Sens.* **10**, DOI: 10.3390/rs10020226 (2018).

13. Shrivastava, S., Singh, S. K. & Hooda, D. S. Color sensing and image processing-based automatic soybean plant foliar disease severity detection and estimation. *Multimed. Tools Appl.* **74**, 11467–11484, DOI: 10.1007/s11042-014-2239-0 (2015).

14. Prasad, S., Peddoju, S. K. & Ghosh, D. Multi-resolution mobile vision system for plant leaf disease diagnosis. *Signal, Image Video Process.* **10**, 379–388, DOI: 10.1007/s11760-015-0751-y (2016).

15. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. journal computer vision* **60**, 91–110 (2004).

16. Lindeberg, T. Feature detection with automatic scale selection. *Int. journal computer vision* **30**, 79–116 (1998).

17. Harris, C. G., Stephens, M. *et al.* A combined corner and edge detector. *Alvey vision conference* **15**, 10–5244 (1988).

18. Bay, H., Tuytelaars, T. & Van Gool, L. Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, 404–417 (Springer-Verlag, Berlin, Heidelberg, 2006).

19. Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **110**, 346–359, DOI: 10.1016/j.cviu.2007.09.014 (2008).

20. Lowe, D. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, 1150–1157, DOI: 10.1109/ICCV.1999.790410 (IEEE, 1999).

21. Ledwich, L. & Williams, S. Object recognition from local scale-invariant features. In *Australian conference on robotics and automation*, 1–8 (2004).

22. Mikolajczyk, K. & Schmid, C. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis machine intelligence* **27**, 1615–1630 (2005).

23. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297, DOI: 10.1007/BF00994018 (1995).

24. Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge University Press, 2000).

25. Stoean, C. & Stoean, R. *Support Vector Machines and Evolutionary Algorithms for Classification* (Springer International Publishing, 2014).

26. Lin, W.-C., Tsai, C.-F., Chen, Z.-Y. & Ke, S.-W. Keypoint selection for efficient bag-of-words feature generation and effective image classification. *Inf. Sci.* **329**, 33 – 51, DOI: 10.1016/j.ins.2015.08.021 (2016). Special issue on Discovery Science.

27. Gao, F. *et al.* A novel semisupervised support vector machine classifier based on active learning and context information. *Multidimens. Syst. Signal Process.* **27**, 969–988, DOI: 10.1007/s11045-016-0396-1 (2016).

28. Islam, M., Dinh, A., Wahid, K. & Bhowmik, P. Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*, 1–4 (IEEE, 2017).

29. Chen, P.-H., Fan, R.-E. & Lin, C.-J. A study on SMO-type decomposition methods for support vector machines. *IEEE Transactions on Neural Networks* **17**, 893–908, DOI: 10.1109/tnn.2006.875973 (2006).

30. V., K., Huang, T. M. & Vogt, M. *Support Vector Machines: Theory and Applications In Support Vector Machines: edited by Lipo Wang* (Springer International Publishing, 2005).

31. Fan, R.-E., Chen, P.-H. & Lin, C.-J. Working set selection using second order information for training support vector machines. *J. Mach. Learn. Res.* **6**, 1889–1918 (2005).

32. Plant village dataset at kaggle. https://www.kaggle.com/emmarex/plantdisease.