Fast and Accurate Detection and Classification of Plant Diseases

H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh

Department of Information Technology

Al-Balqa' Applied University, Salt Campus, Jordan

ABSTRACT

We propose and experimentally evaluate a software solution for automatic detection and classification of plant leaf diseases. The proposed solution is an improvement to the solution proposed in [1] as it provides faster and more accurate solution. The developed processing scheme consists of four main phases as in [1]. The following two steps are added successively after the segmentation phase. In the first step we identify the mostly-green colored pixels. Next, these pixels are masked based on specific threshold values that are computed using Otsu's method, then those mostly green pixels are masked. The other additional step is that the pixels with zeros red, green and blue values and the pixels on the boundaries of the infected cluster (object) were completely removed. The experimental results demonstrate that the proposed technique is a robust technique for the detection of plant leaves diseases. The developed algorithm's efficiency can successfully detect and classify the examined diseases with a precision between 83% and 94%, and can achieve 20% speedup over the approach proposed in [1].

General Terms

Artificial Intelligence, Image Processing.

Keywords

K-means, SGDM Matrix, Color Co-occurrence Method, HSI, Neural Networks.

1. INTRODUCTION

Plant diseases have turned into a dilemma as it can cause significant reduction in both quality and quantity of agricultural products [20]. It is estimated that 2007 plant disease losses in Georgia (USA) is approximately $539.74 million. Of this amount, around 185 million USD was spent on controlling the diseases, and the rest is the value of damage caused by the diseases. Those numbers are listed in Table 1.

The naked eye observation of experts is the main approach adopted in practice for detection and identification of plant diseases [20]. However, this requires continuous monitoring of experts which might be prohibitively expensive in large farms. Further, in some developing countries, farmers may have to go long distances to contact experts, this makes consulting experts too expensive and time consuming [14; 5; 8].

Automatic detection of plant diseases is an essential research topic as it may prove benefits in monitoring large fields of crops, and thus automatically detect the symptoms of diseases as soon as they appear on plant leaves [1; 18; 8]. Therefore; looking for fast, automatic, less expensive and accurate method to detect plant disease cases is of great realistic significance [14; 5].

Machine learning based detection and recognition of plant diseases can provide clues to identify and treat the diseases in its early stages [8; 18]. Comparatively, visually identifying plant diseases is expensive, inefficient, and difficult. Also, it requires the expertise of trained botanist [1].

In [4] the authors have worked on the development of methods for the automatic classification of leaf diseases based on high resolution multispectral and stereo images. Leaves of sugar beet are used for evaluating their approach. Sugar beet leaves might be infected by several diseases, such as rusts (Uromyces betae), powdery mildew (Erysiphe betae).

In [20], a fast and accurate new method is developed based on computer image processing for grading of plant diseases. For that, leaf region was segmented by using Otsu method [15; 12; 13]. After that the disease spot regions were segmented by using Sobel operator to detect the disease spot edges. Finally, plant diseases are graded by calculating the quotient of disease spot and leaf areas.

Studies show that Machine learning methods can successfully be applied as an efficacious disease detection mechanism. Examples of such machine learning methods that have been applied in agricultural researches; Artificial Neural Networks (ANNs), Decision Trees, K-means, k nearest neighbors, and Support Vector Machines (SVMs). For example, Wang et al. in [19] predicted Phytophthora infestans infection on tomatoes by using ANNs. Also, Camargo and Smith in [5] used SVMs to identify visual symptoms of cotton diseases using SVMs.

Table 1: Summary of total losses due to disease damage and cost of control in Georgia, USA in 2007 [11].

Crop	Value of damage ($ millions)	Cost of control
Apple	0.073	0.267
Blueberry	0.14	2.555
Bunch Grape	0.112	0.27
Corn	12.4	0.5
Cotton	81.7	12.2
Muscadine Grape	0.026	0.096
Ornamental	41.22	21.2
Peach	0.177	3.19
Peanut	58.7	41.2
Pecan	0.64	17.4
Soybean	5.3	1.9
Strawberry	0.32	0.683
Turf	126.6	61.2
Vegetable	18.1	20.6
Wheat	0.99	1.8
Totals	**346.49**	185.06

Three are two main characteristics of plant-disease detection machine-learning methods that must be achieved, they are: speed and accuracy.

In this study an automatic detection and classification of leaf diseases has been proposed, this method is based on K-means as a clustering procedure and ANNs as a classifier tool using

some texture feature set. The proposed method is an improvement of the approach proposed in [1].

As a testbed we use a set of leaves which are taken from Al-Ghor area in Jordan. Our program has been tested on five diseases which affect on the plants; they are: Early scorch, Cottony mold, Ashen mold, late scorch, and tiny whiteness. The proposed framework could successfully detect and classify the tested diseases with precision of more than 94% in average with more than 20% speedup over the presented approach in [1]. The minimum precision value was 84% compared to 80% precision in the previous approach.

In conclusion, the aim of this work is threefold: 1) identifying the infected object(s) based upon K-means clustering procedure; 2) extracting the features set of the infected objects using color co-occurrence methodology for texture analysis; 3) detecting and classifying the type of disease using ANNs, moreover, the presented scheme classifies the plant leaves *and stems* at hand into infected and not-infected classes.

2. THE PROPOSED APPROACH – STEP-BY-STEP DETAILS

The overall concept that is the framework for any vision related algorithm of image classification is almost the same. First, the digital images are acquired from the environment using a digital camera. Then image-processing techniques are applied to the acquired images to extract useful features that are necessary for further analysis. After that, several analytical discriminating techniques are used to classify the images according to the specific problem at hand. Figure 1 depicts the basic procedure of the proposed vision-based detection algorithm in this research.

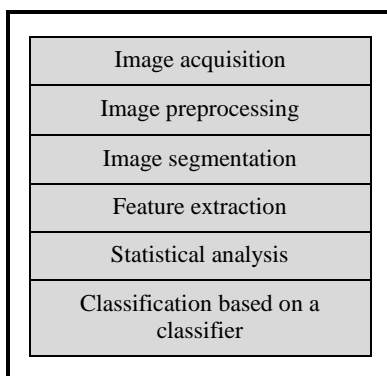| Image acquisition |
|---|
| Image preprocessing |
| Image segmentation |
| Feature extraction |
| Statistical analysis |
| Classification based on a classifier |

Figure 1: The basic procedure of the proposed image-processing-based disease detection solution

The proposed approach step - by - step of the image segmentation and recognition processes is illustrated in Algorithm 1. In the initial step, the RGB images of all the leaf samples were picked up. Some real samples of those diseases are shown in Figure 2. It is obvious from Figure 2 that leaves belonging to early scorch, cottony mold, ashen mold and late scorch have significant differences form greasy spot leaves in terms of *color* and *texture*. Also, Figure 3 shows two images; the left image is infected with tiny whiteness disease, and the right image is a normal image. However, the leaves related to these six classes (early scorch, cottony mold, ashen mold, late scorch, tiny whiteness and normal) had very small differences as discernible to the human eye, which may justify the misclassifications based on naked eye.

Algorithm 1: Basic steps describing the proposed algorithm.

1 RGB image acquisition

2 Create the color transformation structure

3 Convert the color values in RGB to the space specified in the   color transformation structure

4 Apply K-means clustering

5 Masking green-pixels

6 Remove the masked cells inside the boundaries of the infected clusters

7 Convert the infected (cluster / clusters) form RGB to HSI Translation

8 SGDM Matrix Generation for H and S

9 Calling the GLCM function to calculate the features

10 Texture Statistics Computation

11 Configuring Neural Networks for Recognition

Early scorch   Cottony mold

Ashen mold   Late scorch

Figure 2: Sample images from our dataset.

Figure 3: A leaf image infected with tiny whiteness disease (left) and a normal leaf image (right).

Figure 4 shows an example of a plant stem that is infected with white mold disease.

Figure 4: A plant stem that is infected with the white mold disease

In details, in step 2 a color transformation structure for the RGB leaf image is created, and then, a device-independent color space transformation for the color transformation structure is applied in step 3. Steps 2 and 3 are inevitable for carrying out step 4. In this step the images at hand are segmented using the K-Means clustering technique [10; 7; 3; 9]. These four steps constitute phase 1 whereas, the infected object (s) is/are determined. In step 5, we identify the mostly-green colored pixels. After that, based on specified and varying threshold value that is computed for these pixels using *Otsu's method* [12; 13], these mostly green pixels are masked as follows: if the green component of pixel intensities is less than the pre-computed threshold value, the red, green and blue components of the this pixel is assigned to a value of zero. This is done in sense that these pixels have no valuable weight to the disease identification and classification steps, and most probably those pixels represent healthy areas in the leave. Furthermore, the image processing time should become significantly reduced.

In step 6 the pixels with zeros red, green and blue values and the pixels on the boundaries of the infected cluster (object) were completely removed. Steps 5 and 6 form phase 2, and this phase is helpful as it gives more accurate disease classification and identification results with satisfied performance and the overall computation time should become significantly less. The observations behind steps 5 and 6 were experimentally validated.

Next, in step 7 the infected cluster was then converted from RGB format to HSI format. In the next step, the SGDM matrices were then generated for each pixel map of the image for only H and S images. The SGDM is a measure of the probability that a given pixel at one particular gray-level will occur at a distinct distance and orientation angle from another pixel, given that pixel has a second particular gray-level. From the SGDM matrices, the texture statistics for each image were generated. Concisely, the features set were computed only to pixels inside the boundary of the infected areas of the leaf. In other words, healthy areas inside the infected areas were also removed. Steps 7 – 10 form phase 3 in which the texture features for *the segmented infected objects* in this phase are calculated. Finally, the recognition process in the fourth phase was performed to the extracted features through a pre-trained neural network. For each image in the data set the subsequent steps in Algorithm 1 were repeated.

The Proposed approach for segmentation and classification plant diseases can be divided into four phases:

2.1 Phase 1 - K-means Clustering Technique

There are two preprocessing steps that are needed in order to implement the K-means clustering algorithm: The phase starts first by creating device-independent color space transformation structure. In a device independent color space, the coordinates used to specify the color will produce the same color regardless

of the device used to draw it. Thus, we created the color transformation structure that defines the color space conversion. Then, we applied the device-independent color space transformation, which converts the color values in the image to the color space specified in the color transformation structure. The color transformation structure specifies various parameters of the transformation. A *device dependent color space* is the one where the resultant color depends on the equipment used to produce it. For example the color produced using pixel with a given RGB values will be altered as the brightness and contrast on the display device used. Thus the RGB system is a color space that is dependent.

The K-means clustering algorithm tries to classify *objects* (pixels in our case) based on a set of features into K number of classes. The classification is done by minimizing the *sum of squares* of distances between the objects and the corresponding cluster or class *centroid* [10; 7]. However, K-means clustering is used to partition the leaf image into four clusters in which one or more clusters contain the disease in case when the leaf is infected by more than one disease. In our experiments multiple values of number of clusters have been tested. Best results were observed when the number of clusters was 3 or 4. A stem image infected with early scorch and its first cluster (the infected object itself) is shown in Figure 5.

a)   b)

Figure 5: A stem image infected with early scorch; a) original image b) cluster 1 image

An example of the output of K-Means clustering for a leaf infected with *early scorch* disease is shown in Figure 6. It is observed from Figure 6 that cluster 4 contains infected object of early scorch disease. Furthermore, clusters 1 and 2 contain the intact parts of leaf, although they are distinct from each other. However, cluster 3 represents the black background of the leaf which can be discarded primarily. Finally, the image in (e) facilitates the segmentation procedure followed in K-means algorithm.

2.2 Phase 2 – Masking the green pixels and the pixels on the boundaries

This phase consists of two steps: The mostly green colored pixels are identified, and then the global image threshold using *Otsu's method* [12; 13] has been applied in order to specify the varying threshold value which chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels. Next, the green pixels are masked as follows: if the green component of pixel intensities is less than the computed threshold value, then, the red, green and blue components of the this pixel are cleared. The next step in this phase is focused on deleting both the pixels with zeros components and the pixels on the boundaries of the infected cluster(s).

2.3 Phase 3 – Features Extraction

In the proposed approach, the method adopted for extracting the feature set is called the *Color Co-occurrence Method* or

CCM method in short. It is a method, in which both the color and texture of an image are taken into account, to arrive at unique features, which represent that image.

(a)   (b)

(c)   (d)
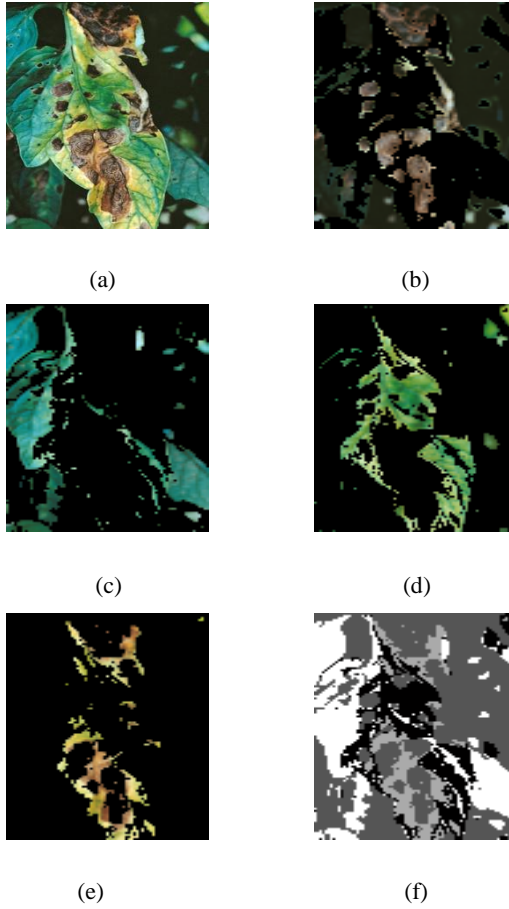
(e)   (f)

Figure 6: An example of the output of K-Means clustering for a leaf that is infected with early scorch disease. (a) The infected leaf picture. (b, c, d, e) the pixels of the first, second, the third and the fourth clusters. (e) a single gray-scale image with the pixels colored based on their cluster index.

2.3.1 Co-occurrence Methodology for Texture Analysis

The image analysis technique selected for this study was the CCM method. The use of color image features in the visible light spectrum provides additional image characteristic features over the traditional gray-scale representation [2].

The CCM methodology established in this work consists of three major mathematical processes. First, the RGB images of leaves are converted into Hue Saturation Intensity (HSI) color space representation. Once this process is completed, each pixel map is used to generate a color co-occurrence matrix, resulting in three CCM matrices, one for each of the H, S and I pixel maps. (HSI) space is also a popular color space because it is based on human color perception [17]. Electromagnetic radiation in the range of wavelengths of about 400 to 700 nanometers is called visible light because the human visual system is sensitive to this range. Hue is generally related to the wavelength of a light and intensity shows the amplitude of a light. Lastly, saturation is a component that measures the "colorfulness" in HSI space [17].

Color spaces can be transformed from one space to another easily. In our experiments, the Equations 1, 2 and 3 were used to transform the image's components from RGB to HSI [6]:

$$Hue\ (H) = \begin{cases} 2 - ACOS\left\{\dfrac{[(R-G)+(R-B)]}{2\sqrt{(R-G)^2+(R-G)(G-B)}}\right\} &, B > G \\[2ex] ACOS\left\{\dfrac{[(R-G)+(R-B)]}{2\sqrt{(R-G)^2+(R-G)(G-B)}}\right\} &, B \le G \end{cases} \quad (1)$$

$$Saturation\ (S) = 1 - \frac{3.\min\ (R,G,B)}{(R+G+B)} \quad (2)$$

$$Intensity\ (I) = \frac{R+G+B}{3} \quad (3)$$

The color co-occurrence texture analysis method was developed through the use of Spatial Gray-level Dependence Matrices (SGDM's) [20]. The gray level co-occurrence methodology is a statistical way to describe shape by statistically sampling the way certain grey-levels occur in relation to other grey-levels.

These matrices measure the probability that a pixel at one particular gray level will occur at a distinct distance and orientation from any pixel given that pixel has a second particular gray level. For a position operator p, we can define a matrix *Pij* that counts the number of times a pixel with grey-level *i* occurs at position *p* from a pixel with grey-level *j*. The SGDMs are represented by the function *P(i, j, d, Θ)* where *i* represents the gray level of the location *(x, y)* in the image *I(x, y)*, and *j* represents the gray level of the pixel at a distance *d* from location *(x, y)* at an orientation angle of *Θ*. The reference pixel at image position *(x, y)* is shown as an asterix. All the neighbors from 1 to 8 are numbered in a clockwise direction. Neighbors 1 and 5 are located on the same plane at a distance of *1* and an orientation of *0* degrees. An example image matrix and its SGDM are already given in the three equations above. In this research, a *one* pixel offset distance and a *zero* degree orientation angle was used.

After the transformation processes, we calculated the feature set for H and S, we dropped (I) since it does not give extra information. However, we use GLCM function in Matlab to create gray-level co-occurrence matrix; the number of gray levels is set to 8, and the symmetric value is set to "true", and finally, offset is given a" 0" value.

2.3.2 Normalizing the CCM matrices

The CCM matrices are then normalized using Equation 4.

$$p(i,j) = \frac{p(i,j,1,0)}{\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}p(i,j,1,0)} \quad (4)$$

$p(i,j)$ is the image attribute matrix, $p(i,j,1,0)$ represents the intensity co-occurrence matrix, and N_g represents the total number of intensity levels.

The marginal probability matrix (p_x) can be defined as shown in Equation 5.

$$p_x(i) = \sum_{j=0}^{N_g-1} p(i,j) \quad (5)$$

Sum and difference matrices (p_{x+y}, p_{x-y}) are defined as shown in Equations 6 and 7, respectively.

$$p_{x+y}(k) = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} p(i,j) \quad (6)$$

Where $k = I + j$; for $k = 0, 1, 2, ... 2(N_g - 1)$,

$$p_{x-y}(k) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j) \qquad (7)$$

Where $k = |I - j|$; for $k = 0, 1, 2, \ldots 2(N_g - 1)$.

2.3 Texture Features Identification

The following features set were computed for the components H and S:

- The angular moment (E) is used to measure the homogeneity of the image, and is defined as shown in Equation 8.

$$E = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} [p(i,j)]^2 \qquad (8)$$

- The produc moment (cov) is analogous to the covariance of the intensity co-occurance matrix and is defined as shown in Equation 9.

$$cov = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - I_2)(j - I_2)P(i,j) \qquad (9)$$

- The sum and difference entropies (se and de) which are computed using Equations 10 and 11 respectively.

$$se = \sum_{k=0}^{2(N_g-1)} P_{x+y}(k)\, lnP_{x+y}(k) \qquad (10)$$

$$de = \sum_{k=0}^{N_g-1} P_{x-y}(k)\, lnP_{x-y}(k) \qquad (11)$$

- The entropy feature (e) is a measure of the amount of order in an image, and is computed as as defined in Equation 12.

$$e = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j)\, lnP(i,j) \qquad (12)$$

- The information measures of correlation ($inf2h$). $inf2h$ is defined as shown in Equation 13.

$$inf2h = \left[1 - e^{-2(Hxy-e)}\right]^{1/2}. \qquad (13)$$

Where:

$$Hxy = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P_x(i)P_x(j)\, ln[P_x(i)P_x(j)]$$

- Contrast (id) of an image can be measured by the inverse difference moment as shown in Equation 14.

$$id = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{P(i,j)}{1+(i-j)^2} \qquad (14)$$

- Correlation (c) is a measure of intensity linear dependence in the image and is defined as shown in Equation 15.

$$c = \frac{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} ijP(i,j) - I_2^2}{I_3} \qquad (15)$$

2.4 Phase 4 – Neural Networks

In this paper, neural networks are used in the automatic detection of leaves diseases. Neural network is chosen as a classification tool due to its well known technique as a successful classifier for many real applications. The training and validation processes are among the important steps in developing an accurate process model using NNs. The dataset for training and validation processes consists of two parts; the training feature set which are used to train the NN model; whilst a testing features sets are used to verify the accuracy of the trained NN model.

Before the data can be fed to the ANN model, the proper network design must be set up, including type of the network and method of training. This was followed by the optimal parameter selection phase. However, this phase was carried out simultaneously with the network training phase, in which the network was trained using the feed-forward back propagation network. In the training phase, connection weights were always updated until they reached the defined iteration number or acceptable error. Hence, the capability of ANN model to respond accurately was assured using the Mean Square Error (MSE) criterion to emphasis the model validity between the target and the network output.

3. EXPERIMENTAL RESULTS AND OBSERVATIONS

3.1 Input Data Preparation and Experimental Settings

In our experiments, two main files were generated, namely: (i) Training texture feature data, and (ii) Testing texture feature data. The two files had 192 rows each, representing 32 samples from each of the six classes of leaves. Each row had 10 columns representing the 10 texture features extracted for a particular sample image. Each row had a unique number (1, 2, 3, 4, 5 or 6) which represented the class (i.e., the disease) of the particular row of data. '1' represented early scorch disease infected leaf. '2' represented Cottony mold disease infected leaf. '3' represented ashen mold disease infected leaf. '4' represented late scorch disease infected leaf. '5' represented tiny whiteness disease infected leaf, and '6' represented normal leaf. Then, a software program was written in MATLAB that would take in **.mat** files representing the training and testing data, train the classifier using the 'train files', and then use the 'test file' to perform the classification task on the test data. Consequently, a Matlab routine would load all the data files (training and testing data files) and make modifications to the data according to the proposed model chosen. In the experimental results, the threshold value for each of the above categories is constant for all samples infected with the same disease. This threshold is a global image threshold that is computed using Otsu's method [12; 13].

The architecture of the network used in this study was as follows. A set of 10 hidden layers in the neural network was used with the number of inputs to the neural network (i.e. the number of neurons) is equal to the number of texture features listed above. The number of output is 6 which is the number of classes representing the 5 diseases studied along with the case of normal (uninfected) leaf. Those diseases are early scorch, cottony mold, ashen mold, late scorch, tiny whiteness. The neural network used is the feed forward back propagation with the performance function being the Mean Square Error (MSE) and the number of iterations was 10000 and the maximum allowed error was 10^{-5}.

3.2 Experimental Results

The results for NN classification strategy for testing samples were given in Table 2. These results were obtained using a NN classifier for different five diseases. In particular, model M1 achieved the highest overall classification accuracy, in which it achieved an overall accuracy of 94% compared to the 89.5% accuracy achieved in [1]. Also, it can be concluded that model M1 is the best overall model in this classifier in terms of accuracy and in computational time for both training and classification. In conclusion, Table 2 reported better classification accuracies for all the data models. Also, Figure 7

shows a graph that representing the percentage classification of various diseases of all the data models shown in Table 2.

Table 2: Percentage classification of various diseases

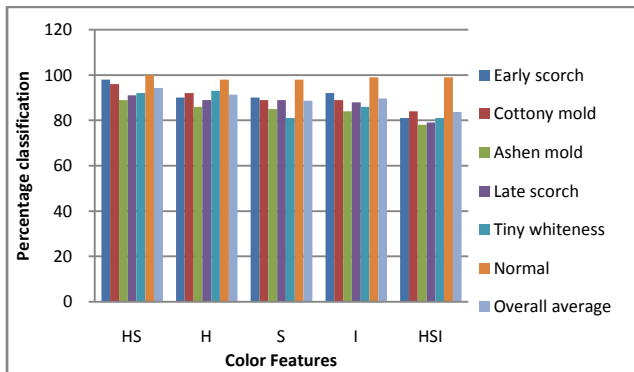| Model	Color Features used	Early scorch	Cottony mold	Ashen mold	Late scorch	Tiny whiteness	Normal	Overall average
M1	HS	98	96	89	91	92	100	94.33
M2	H	90	92	86	89	93	98	91.33
M3	S	90	89	85	89	81	98	88.67
M4	I	92	89	84	88	86	99	89.67
M5	HSI	81	84	78	79	81	99	83.67

Figure 7: Percentage classification of various diseases

The recognition rate for NN classification strategy of all models was also computed for early scorch, cottony mold and normal leaf images based upon Algorithm 1, the obtained results for Model 1 and 5 are reported in Table 3.

Table 3: Recognition rates of individual plant diseases

Model	Color Features	Early scorch	Cottony mold	Normal	Overall average
M1	HS	99	100	100	99.66
M5	HSI	96	98	100	98.00

It can be implied from Table 3 that Model M1 which has used only the H and S components in computing the texture features, has emerged as the best model among the various models. Furthermore, it can be observed from Table 3 that Model M5 has less recognition rate than Model 1; this was in part because of the elimination of the intensity (I component) from computing the texture features in Model M1. As a matter of fact, elimination of intensity is serviceable in this study because it nullifies the effect of intensity variations. The numbers of leaf samples that were classified into each of the five tested categories using model M1 with specific threshold value are shown in Table 4 and Figure 8. It is observed from Table 4 that only few samples from late scorch and tiny whiteness leaves were misclassified, also, three test images were misclassified for the case of late scorch infected leaves. Similarly, in the case of tiny whiteness images, only two test images from the class were misclassified. In average, accuracy of classification using our approach was 94.67 compared to 92.7 in case of using the approach presented in [1].

Table 4: Classification results per class for neural network with back propagation.

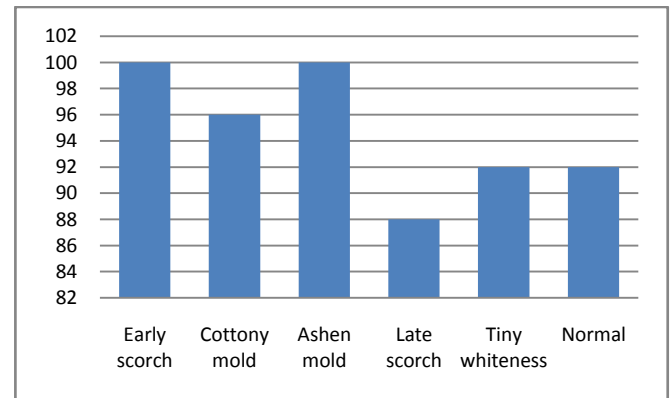| From species	Early scorch	Cottony mold	Ashen mold	Late scorch	Tiny whiteness	Normal	Accuracy
Early scorch	25	0	0	0	0	1	100
Cottony mold	0	24	0	1	0	0	96
Ashen mold	0	0	25	0	0	1	100
Late scorch	0	0	0	22	1	0	88
Tiny whiteness	0	1	0	0	23	0	92
Normal	0	0	0	2	1	23	92
Average							**94.67**

Figure 8: Classification results per class for neural network with back propagation.

The convergence curve for the learning step of the neural network in the proposed study in this work is better than that of [1] as shown in Figure 9. In Figure 9 Approach 1 represents the study presented in [1], while Approach 2 represents our proposed study.
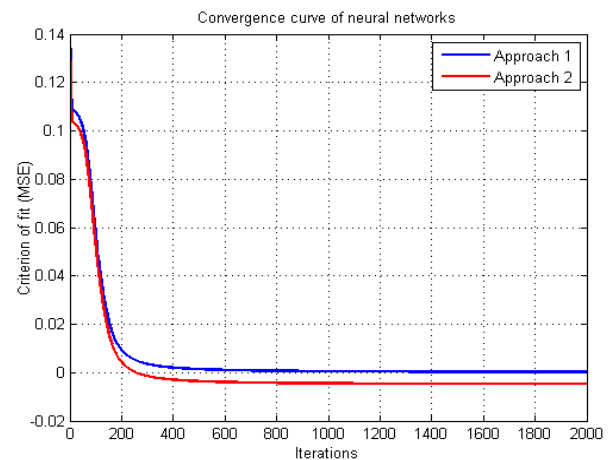

Figure 9: The convergence curve of [1], that is "Approach 1", and our approach "Approach 2".

It can be concluded from the above tables and figures that the obtained results achieved an acceptable level of optimal results which can also add more weighting rate to the proposed study. The presented approaches in this paper and in [1] have been implemented in MATLAB under Windows XP environment. All the experiments are carried out on a desktop computer with Intel (R) Pentium (R) CPU 2.20GHZ 645 MHZ and 256MB of RAM. The average computation time for the two proposed approaches was computed in seconds for the models M1, M2, M3, M4 and M5 as shown in Table 5. The data in Table 5 was obtained for the two approaches using the same neural network structure and under the same machine. Figure 10 shows that our proposed approach has 19% speedup over the approach of [1].

Table 5: Average computation time.

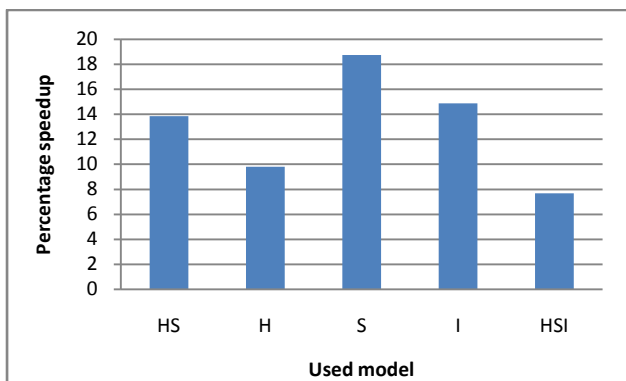| Model	Approach 1	Approach 2
M1	456.19	392.98
M2	399.00	359.91
M3	477.91	388.27
M4	408.22	347.44
M5	449.12	414.60
Average	**438.09**	**380.64**

Figure 10: Speedup of computation times (approach 2 over approach 1).

4. CONCLUSION AND FUTURE WORK

In this paper, respectively, the applications of K-means clustering and Neural Networks (NNs) have been formulated for clustering and classification of diseases that affect on plant leaves. Recognizing the disease is mainly the purpose of the proposed approach. Thus, the proposed Algorithm was tested on five diseases which influence on the plants; they are: Early scorch, Cottony mold, ashen mold, late scorch, tiny whiteness.

The experimental results indicate that the proposed approach is a valuable approach, which can significantly support an accurate detection of leaf diseases in a little computational effort.

An extension of this work will focus on developing hybrid algorithms such as genetic algorithms and NNs in order to increase the recognition rate of the final classification process underscoring the advantages of hybrid algorithms; also, we will dedicate our future works on automatically estimating the severity of the detected disease.

REFERENCES

[1]. Al-Bashish, D., M. Braik and S. Bani-Ahmad, 2011. Detection and classification of leaf diseases using K-means-based segmentation and neural-networks-based classification. Inform. Technol. J., 10: 267-275. DOI: 10.3923/itj.2011.267.275

[2]. Aldrich, B.; Desai, M. (1994) "Application of spatial grey level dependence methods to digitized mammograms," Image Analysis and Interpretation, 1994., Proceedings of the IEEE Southwest Symposium on, vol., no., pp.100-105, 21-24 Apr 1994. DOI: 10.1109/IAI.1994.336675

[3]. Ali, S. A., Sulaiman, N., Mustapha, A. and Mustapha, N., (2009). K-means clustering to improve the accuracy of decision tree response classification. Inform. Technol. J., 8: 1256-1262. DOI: 10.3923/itj.2009.1256.1262

[4]. Bauer, S. D.; Korc, F., Förstner, W. (2009): Investigation into the classification of diseases of sugar beet leaves using multispectral images. In: E.J. van Henten, D. Goense and C. Lokhorst: Precision agriculture '09. Wageningen Academic Publishers, p. 229-238. URL: http://www.precision-crop-protection.uni-bonn.de/gk_research/project.php?project=3_09

[5]. Camargo, A. and Smith, J. S., (2009). An image-processing based algorithm to automatically identify plant disease visual symptoms, Biosystems Engineering, Volume 102, Issue 1, January 2009, Pages 9-21, ISSN 1537-5110, DOI: 10.1016/j.biosystemseng.2008.09.030.

[6]. Ford, A. and Roberts, A., (2010) Colour Space Conversions, August 11, 1998. http://www.poynton.com/PDFs/coloureq.pdf, viewed on May 2010.

[7]. Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C (Applied Statistics) 28 (1): 100–108.

[8]. Hillnhuetter, C. and A.-K. Mahlein, Early detection and localisation of sugar beet diseases: new approaches, Gesunde Pfianzen 60 (4) (2008), pp. 143–149.

[9]. Jun, W. and S. Wang, (2008). Image thresholding using weighted parzen-window estimation. J. Applied Sci., 8: 772-779. DOI: 10.3923/jas.2008.772.779. URL: http://scialert.net/abstract/?doi=jas.2008.772.779

[10]. MacQueen, J.. (1967). Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman, editors, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281--297, Berkeley, CA, 1967. University of California Press. URL: http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.bsmsp/1200512992.

[11]. Martinez, A., (2007). 2007 Georgia Plant Disease Loss Estimates. http://www.caes.uga.edu/Publications/displayHTML.cfm?pk_id=7762. Viewed on Saturday, 15, January, 2011.

[12]. Otsu, N. (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9: 62–66. DOI:10.1109/TSMC.1979.4310076.

[13]. Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[14]. Prasad Babu, M. S. and Srinivasa Rao , B. (2010) Leaves recognition using back-propagation neural network - advice for pest and disease control on crops. Technical report, Department of Computer Science & Systems Engineering, Andhra University, India. Downloaded from www.indiakisan.net on May 2010.

[15]. Sezgin, M. and Sankur, B. (2003). "Survey over image thresholding techniques and quantitative performance evaluation". Journal of Electronic Imaging 13 (1): 146–165. DOI:10.1117/1.1631315.

[16]. Soltanizadeh, H. and B.S. Shahriar, 2008. Feature extraction and classification of objects in the rosette pattern using component analysis and neural network. J. Applied Sci., 8: 4088-4096. DOI: 10.3923/jas.2008.4088.4096. URL: http://scialert.net/abstract/?doi=jas.2008.4088.4096.

[17]. Stone, M. C. (August 2001). "A Survey of Color for Computer Graphics". Course at SIGGRAPH 2001.

[18]. Rumpf, T., A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, L. Plumer, Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance, Computers and Electronics in Agriculture, Volume 74, Issue 1, October 2010, Pages 91-99, ISSN 0168-1699, DOI: 10.1016/j.compag.2010.06.009.

[19]. Wang, X., M. Zhang, J. Zhu and S. Geng, Spectral prediction of Phytophthora infestans infection on tomatoes using artificial neural network (ANN), International Journal of Remote Sensing 29 (6) (2008), pp. 1693–1706.

[20]. Weizheng, S., Yachun, W., Zhanliang, C., and Hongda, W. (2008). Grading Method of Leaf Spot Disease Based on Image Processing. In Proceedings of the 2008 international Conference on Computer Science and Software Engineering - Volume 06 (December 12 - 14, 2008). CSSE. IEEE Computer Society, Washington, DC, 491-494. DOI= http://dx.doi.org/10.1109/CSSE. 2008.1649.