Disease Detection and Classification of Citrus leaves using Feature Weighted Mahalanobis and Neural Networks Classifiers

Nikhil Niphadkar[1], Shashikiran Prabhakar[1], and Venkat Jayaraman[1]

[1]Agricultural and Biological Engineering Department, University of Florida, Gainesville, FL 32601, USA

The citrus industry, being an important constituent of Florida's overall agricultural economy postulates proper disease control measures in citrus groves to minimize losses. Melanose, greasy spot and scab are the most devastating diseases that threaten citrus crops. Technologies that can efficiently identify these diseases would assure fruit quality and safety and enhance the competitiveness and profitability of the citrus industry. This project is aimed at investigating the potential of using pattern classification techniques for detecting the disease lesions on citrus leaves. The leaves are classified into four categories based on the disease i.e. scab, melanose, greasy spot and normal leaf. Two classifiers were designed for a comparative study to determine the best approach to achieve the disease classification. Classification of the extracted features, using feature weighted Mahalanobis distance which is a supervised classification algorithm and a neural network classification based on back-propagation algorithm. A study of the two shows that the neural network classifier exhibits better performance as compared to the feature weighted classifier. The study also showed the improved performance of the Feature Weighted Classifier as compared to the Mahalanobis minimum distance criterion.

***Index Terms*— Citrus Disease Detection, Feature Weighted Mahalanobis distance, Neural Networks**

I. INTRODUCTION

CITRUS DISEASES like Melanose, scab and greasy spot are severe diseases that affect most commercial varieties. The pathogen, usually characterized by conspicuous, erumpent lesions on leaves, stems, and fruit, could cause defoliation, blemished fruit, premature fruit drop, twig dieback, and tree decline. Although there are various disorders that could infect citrus leaves, scab, melanose and greasy spot are considered as one of the most devastating diseases that threaten citrus crops. Unfortunately, there is no treatment or prevention at this time that could completely eradicate these diseases in the infected areas (e.g., southern Florida). The current emphasis is put on minimizing the level of the disease and avoiding further infections.

The overall objective of this project is to investigate the potential of different classifiers to distinguish normal leaves from leaves diseased with scab, melanose and greasy spot and present a comparative study of the results from the two classifiers. Using this technology we can efficiently identify these diseases and assure fruit quality and safety and enhance the competitiveness and profitability of the citrus industry.

In the past decade, agricultural applications using image processing and pattern recognition techniques have been attempted by various researchers. Object shape matching functions such as those used by Woebbecke et al. (1995a) to develop a vision system using shape features which included were roundness, aspect, perimeter, thickness, elongatedness and several invariant central moments (ICM) for identifying young weeds. Kataoka et al. (2001), used color-based classifier based on thresholding Munsell and XYZ color systems to develop an automatic detection system for detecting apples ready for harvest. Burks (2000a) developed a method for classification of weed species using color texture features and discriminant analysis. The image analysis technique

adding up of the variance normalized squared distances of the features. But if the feature is distorted by noise, due to the squaring of the distances, a single feature can have such a high value that it covers the information provided by the other features and leads to a misclassification. We implement the Mahalanobis Classifier and improve on this limitation by using feature weighted Mahalanobis distance (Ref: Wolfel and Ekenel et al. [5]) and compare the performance with the neural network classifier.

II. TECHNOLOGY / DATA

A. Image Acquisition

The 3 CCD camera (JAI, MV90) used for image acquisition are calibrated under the artificial light source using a calibration gray-card. An RGB digital image was taken of the gray-card, and each color channel is evaluated using histograms, mean, and standard deviation statistics. Red and green channel gains are adjusted until the gray-card images having similar means in R, G, and B, equal to approximately 128, which is mid-range for a scale from 0 to 255.

The image acquisition system consists of the following principal components:

- Four 16 W cool white fluorescent bulbs (4500K) with natural light filters and reflectors were located approximately 75 cm above the image plane.
- The JAI MV90, 3 CCD color camera with 28-90 mm zoom lens.
- A Coreco PC-RGB 24-bit color frame grabber with 480 × 640 pixel resolution, with an approximately 7 × 10 cm field of view.
- MV Tools image capture software

B. Feature Extraction

Four different classes of citrus leaves are selected for this study. The diseased leaf samples to be investigated are greasy spot, melanose, scab, and normal citrus leaf. Leaf sample of images are shown in Fig. 1. The degree of disease damage varies between leaf samples and images of these are acquired in controlled lighting conditions using a color CCD camera of 640 x 480 resolution. In the next step, the image size is reduced from 480 × 640 pixel resolution to 240 × 320 pixel resolution. The reduced images are then converted from 8 bits per channel RGB format to 6 bits per channel HSI format.

The Spatial Gray-Level Dependency Matrices (SGDM's) are generated for each color pixel map of the image, one each for hue, saturation and intensity. The experiment uses the 0° CCM orientation angle and an offset distance of one pixel for the finest texture measure. From the SGDM matrices, the 39 CCM texture statistics are generated for each image using the three color feature co-occurrence matrices, as each SGDM matrix provides 13 texture features.(Ref: Shearer[3], Burks [4]).

The hue, saturation, and intensity CCM matrices were then used to generate the texture features described by Haralick and Shanmugam (1974) [7]. Shearer and Holmes (1990) [5] reported a reduction in the 16 gray scale texture features through elimination of redundant variables. The resulting 11 texture feature equations are defined by Shearer and Holmes (1990) and Burks (1997). The same equations are used for each of the three CCM matrices, producing 11 texture features for each HSI component and thereby a total of 33 CCM texture statistics. The image acquisition and CCM texture analysis method is illustrated in a flowchart format in Fig. 2.

Once the Textural statistics were generated, a statistical analysis was conducted using SAS statistical analysis package to reduce redundancy in the texture feature set which acted as the input data to our classifiers. Based on these analyses several models of data sets were created which are shown in Table 3.

III. THEORY/METHODS

Once the textural statistics are generated for each image, the classification will be based on feature weighted Mahalanobis classifier and back-propagation neural network classifier.

A. Feature Weighted Mahalanobis distance Classifier

The Mahalanobis distance is a very useful way of determining the "similarity" of a set of values from an "unknown" sample to a set of values measured from a collection of "known" samples. One of the main reasons the Mahalanobis distance method is used is because it is very sensitive to inter-variable changes in the training data. In addition, since the Mahalanobis distance is measured in terms of standard deviations from the mean of the training samples, the reported matching values give a statistical measure of how well the spectrum of the unknown sample matches (or does not match) the original training spectra. This method belongs to the class of supervised classification algorithms. Earlier research by Shearer (1986) had shown that plant canopy texture features can be represented by a multi-variate normal distribution. The feature space can be approximated to be a mixed Gaussian model containing a combination of 39 univariate normal distributions, if all the features are considered. The feature space is a mixture of Gaussian models containing a combination of N univariate normal distributions, where N is the number of texture features in the model. Once it is known that the feature space of various classes of leaves is a mixed Gaussian model, the next step is to calculate the statistics representing those classes. . Four parameter sets $X[(\mu,\Sigma)]$ (mean and covariance), representing the four classes of diseased and normal leaves (greasy spot, melanose, normal, and scab) are calculated using the training images. This stage represented the training phase, wherein the necessary statistical features representing various classes of leaves are calculated.

After the training parameter sets are obtained, the classifier is evaluated using the test images for each class. This will constitute the validation phase. The classifier is based on the squared Mahalanobis distance from the feature vector representing the test image to the parameter sets of the various classes. It uses the nearest neighbor principle. The formula for calculating the squared Mahalanobis distance metric (r) is:

$$r^2 = (x-\mu)^T \Sigma^{-1} (x-\mu)$$

x = N-dimensional test feature vector (N is the number of features considered)
μ = N-dimensional mean vector for a particular class of leaves
Σ = N × N dimensional covariance matrix for a particular class of leaves.
During the testing phase, the squared Mahalanobis distance, for a particular test vector representing a leaf, was calculated for all the classes of leaves. The test image was then classified using a Gaussian classifier based on the Bayes decision rule $i_{Bayes} = \arg\max_i P(\omega_i \mid x)$

$$p(x \mid \omega_i)$$

Using a Gaussian classifier (GC) uses Bayes $i_{Bayes} = \arg\max_i (-\frac{1}{2} D_i + \log P(\omega_i))$ decision theory where the class-conditional probability density $p(x \mid \omega_i)$ is assumed to have a Gaussian distribution for each class ω_i

Under the assumption of multivariate Gaussian densities the discriminant function can be written as:

$$i_{Bayes} = \arg\max_i (-\frac{1}{2} D_i + \log P(\omega_i)) \quad (1)$$

The test image was classified as belonging to a particular class to which its discriminant was the argmax of the calculated distances. To improve the robustness of this classifier we use feature weighted Mahalanobis distance (Ref: Wolfel and Ekenel et al. [8])

1. Feature Weighted Mahalanobis Distance.

The features which are distorted by noise have, in average, a higher influence on the distance measure than the less distorted features as they are further away from the feature mean of the class. Therefore, we lower the influence of these features by reducing their weight. To find the features which have the strongest influence on the distance we solve the Mahalanobis distance equation for every single feature *c* over all input samples *i* and classes *j* and store the value in *Z*

$$\forall c, i, j : Z_{i,j}[c] = (x_i[c] - \mu_j[c])\Sigma_j[c,c]^{-1}(x_i[c] - \mu_j[c])$$

Online training is used in the project, in which one data point is being fed to the network at a time, its output is generated and is compared with the target output (i.e. class label) to calculate the errors in the previous stage (delta values) [9], which is finally used to calculate the error gradient during back propagation. In the final stage of training, the weights are updated using the error gradient along with the regularization constant. As the sequential training converges faster than the batch training, it is being used in the project. To limit the complexity of the network we have implemented a network with only a single hidden layer.

The Logistic Sigmoid functions are used as the activation functions. We have used the number of output nodes equal to the number of classes, each of which corresponds to one of the classes. The node, which gives the maximum value, is labeled as the class label.

After the network is trained and weights are being updated, the network is run on a separate validation dataset and the performance is measured by plotting the error plots.

The algorithm and terminology mentioned in Bishop [9] is followed in the project.

2. *Experiments:*

Initially the network performance is checked on the simple datasets like cross, ellipse, (binary classification) and iris (three classes) which were taken from internet [10]. After the satisfactory performance of the network on these simple datasets, the trials on actual leaves' data are carried out.

3. *Cross validation:*

A four fold cross validation was performed to train the model. The dataset was divided into four subsets and the network was trained on three of the subsets and tested on remaining one. After getting good results by this method, training was done on three of the above mentioned subsets and the testing was done on the entire training dataset. In the end the testing was carried out on a separate validation dataset. Before training on any of the folds, the weights are re-initialized randomly. This ensured that the training was totally independent of random initialization of the weights. The best network was selected out of the four networks depending upon which of them was giving minimum error on the validation set.

4. *Reduced features selection:*

The network performance is verified on the reduced datasets (by reducing the number of features) which are extracted from the actual 39 featured dataset. Four models mentioned by Pydipati (2005) were used. The models were obtained using Principal component Analysis.

The following experiments were carried out on the datasets to judge the performances of each classification models.

1. Checking the effect of changing the learning rate on the behavior of the network.
2. Checking the effect of changing the number of hidden nodes on Multilayer Perceptron.
3. Effect of Regularization term in case of over fitting.
4. Checking the effect of imbalance of the data points. (if majority of points are in one class)
5. Checking the effect of number of iterations.
6. Extraction of the confusion matrix.

IV. RESULTS AND OBSERVATIONS:

A. Neural Network Classifier

The data in Table 1 represents the best results obtained on different models. And as can be observed from the confusion matrix the testing is done on a subset of training data (20 data points out of 80) during cross validation.

The different models of training contain the following features.
Train_1F = S2, H10, H6, H2, H8, S9, S4

Train_2F = I2, I5, I4, I12, I13
Train_3F = I2, I5, I4, S2, H11, S4, H4
Train_new = All 39 features
Test_1F = S2, H10, H6, H2, H8, S9, S4
(Separate validation set)

Note: The separate validation set also has 39 features but as mentioned later, we obtained good results on Train_1F model and hence separate dataset with the same features as Train_1F model is mentioned here.

The observations and results are tabulated in Table 1. The following are the observations that can be derived:

- Best results are obtained from dataset Train_1F
- As number of Hidden units decreases percentage error in training and validation increases.
- Training error as less as 8 % is obtained with lesser hidden units (n = 17) however there are huge oscillations in the error plot as can be seen from the Fig 4 as against the higher number of hidden units (n = 50) Fig 5
- Increasing the number of hidden units consistently decreases the training error with minimal oscillations and we can obtain very small error.
- Increasing the number of hidden units (101) reduces the error however the computation time increases as the number of hidden units increase.
- The optimal compromise between number of hidden units and computation time was achieved with hidden units around 40 to 60.
- As can be seen from the confusion matrix of the Train_1F model (Table 1), we have achieved 0% testing error during classification. We usually obtain the training error of 1.67 to 8.33 %. The confusion matrix with zero error is deliberately displayed here just to highlight that this may not be the true picture of the working model. As more and more data points are observed the error rate may increase. This can be observed from the confusion matrix in Table 2 in which the network is trained on three subsets of the training data and tested on all of it (i.e. tested on whole dataset of 80 data points).
- Increasing the learning rate (LR) increases the classification accuracy however if we increase it too much the performance of the network deteriorates.
- The values of LR in the range 0.01 to 0.04 gave us acceptable results but the best results are obtained by using the LR in the range 0.01 to 0.025. If we try to increase LR more than 0.04 the misclassification rate increases.
- Increasing number of iterations yields better results.
- Large number of iteration say 25000, results in negligible errors which could be a result of over training or over fitting. The regularization constant (λ) helped control these phenomena of over fitting.
- For the given problem complexity (number of data points) it was observed that the over fitting problem becomes less severe if the size of the dataset is increased. This was achieved by adding same data points twice or thrice and creating a double or triple large dataset and then feeding it to the network.
- Also while doubling or tripling the data we added more number of the data points of a same class and deliberately making the dataset biased to one class. This resulted in training of the model more on one class and hence while testing, network misclassified most of the data points.
- Excessively high values of regularization constant (λ) gave poor results. Also the training as well as testing error seems to increase largely due to higher values of regularization constant.
- After considerable number of trials, acceptable range for the regularization constant (λ) is 0.0001 to 0.005 with the optimal value of 0.001

Conclusion:

The following conclusions can be made from implementation of the MLP algorithm

- Error Back Propagation excessive learning can cause divergence of learning.
- Training error initially reduces over the number of iteration and after certain point remains constant.
- Testing error initially decreases and then starts growing.
- Increasing the number of hidden units or hidden layers does not always give good result instead it may result into over fitting.
- The regularization term controls the over fitting and hence restricts the testing error from increasing.
- If the dataset is biased to one of the classes, network misclassifies majority of data points and the performance of the network is difficult to improve as the dataset itself is faulty.

B. Feature Weighted Mahalanobis Classifier

It is observed from the results that the Feature weighted Mahalanobis distance significantly reduced the classification error in the Gaussian classifier and that the Difference Weighted Distance exhibits better performance compared to the Descent Weighted Distance.

1 Validation.

The datasets were varied as per the models mentioned in Table 3 and the results for all the three distances were plotted subsequently in tables 4, 5 and 6. The results shown in Table 4 were obtained using the Mahalanobis Distance in the Gaussian classifier. In particular, model 1B achieved better overall classification rates. Model 1B achieved an overall accuracy of 76.25 %.

By weighing the Mahalanobis distance using Feature Weighted approach, the classification accuracy showed an improvement as seen in Table 5 and Table 6.

The Descent feature weighted approach resulted in an overall accuracy of 81.25% for model 1B which is a significant increase from the un-weighted approach.

The best result was observed in Difference Feature weighted approach where the accuracy for models 1B and 4B exhibited 90% and 91.25% accuracy which indicate the presence of some noisy features which have been weighted down by this process and

2 Conclusion

The performance of the Gaussian Classifier based on Mahalanobis distance can be enhanced by using feature weights which reduce or increase the influence of the features based on the method. It can be concluded the statistical classification It is evident from Tables 4, 5 and 6 that, for models 2B and 3B, the classification accuracies for some classes of leaves were inconsistent. Mahalanobis distance method determines the similarity of a set of values from an unknown sample (test data) to a set of values measured from a collection of known samples (training data). For models 2B and 3B, it may be the case that the spectrums for training and test were dissimilar for the particular choice of texture features in those models. Since the classification was predominantly based on the Mahalanobis distance, even a slight off bound may significantly affect the test accuracies. Therefore, the choice of the method, model and hence the texture features selected, significantly affect the classification results.

C. Future Scope

There are a lot of avenues for improving the classification, especially using the Feature Weighted method. A comparative study of the different classification methods can be used as a valuable tool in deducing the ideal methods for the disease classification in the leaves. Feature Weights can be concatenated to obtain better results especially in the presence of noise. The performance variation and accuracy in the presence of noise and missing features can be conducted to determine the robustness of the classifiers along with different feature reducing techniques such as linear discriminant analysis.

REFERENCES

[1] Burks, T. F., S. A. Shearer, and F. A. Payne. 2000. Classification of weed species using color texture features and discriminant analysis. *Trans. ASAE* 43(2): 441-448.

[2] Nakano, K. 1997. Application of neural networks to the color grading of apples, *Computers and Electronics in Agric.* 18(2-3):105-116.

[3] Shearer, S. A. 1986. Plant identification using color co-occurrence matrices derived from digitized images, PhD diss. Columbus, Ohio: Ohio State University

[4] TF Burks, SA Shearer, JD Green, JR Heath. 2002. Influence of weed maturity levels on species classification using machine vision, *Weed Science,* 50:802–811. 2002

[5] Shearer, S. A., and R. G. Holmes. 1990. Plant identification using color co-occurrence matrices. *Transactions of the ASAE* 33(6):2037-2044.

[6] Burks, T. F. 1997. Color image texture analysis and neural network classification of weed species. Ph.D.thesis. Lexington, Ky.:University of Kentucky, Biological and Agricultural Engineering .

[7] Haralick, R. M., and K. Shanmugam. 1974. Combined Spectral and Spatial Processing of ERTS Imagery Data. *J. Remote Sens. Environ.* 3: 3-13.

[8] M Wolfel and H.K.Ekenel, Feature Weighted Mahalanobis Distance Improved Robustness for Gaussian Classifiers, Institut fuer Theoretische Informatik, University at Karlsruhe

[9] Pattern Recognition and Machine Learning by Christopher M. Bishop

[10] Link for the simple datasets like Cross, Ellipse, Iris. http://www.cise.ufl.edu/class/cap6615sp08/Project1/

Fig. 3 Neural Network Back propagation algorithm

FIGURES AND TABLES

Fig 1- (a) Greasy spot (b) Melanose (c) Scab (d) Normal citrus

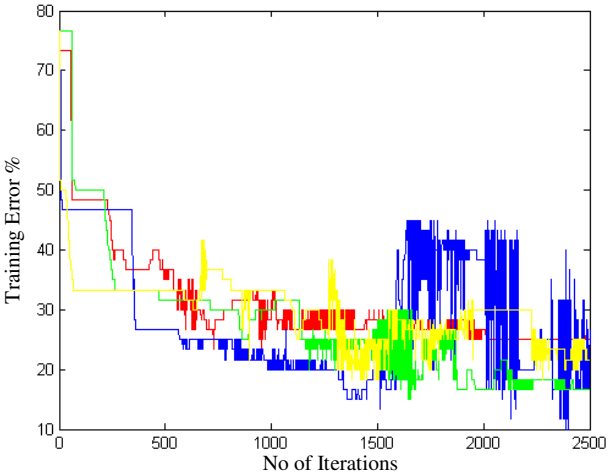


Fig 4: Oscillation in Error Plot due to lesser Hidden Units

Fig. 2 Image Acquisition and Feature extraction system

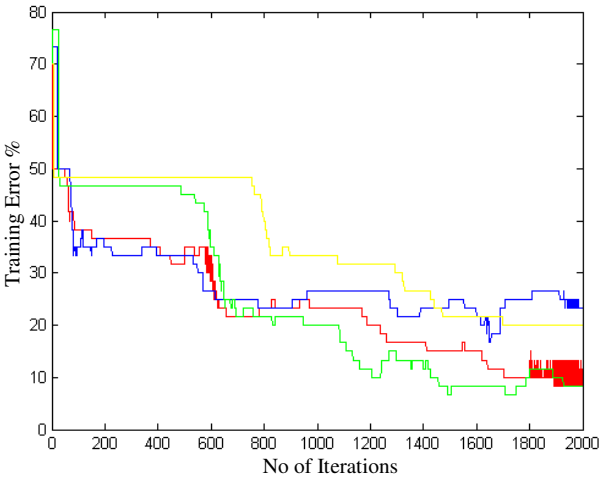
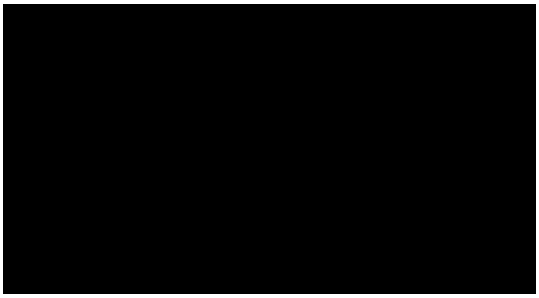


Fig 5: Reduced Oscillations in Error Plot due to higher hidden units



	Train_1F	Train_2F	Train_3F	Train_new
No of Hidden Units	54	47	39	71
Learning rate	0.016	0.020	0.010	0.021
Regularization Constant	0.001	0.001	0.0014	0.002
Minimum % error on Training Data	1.67	16.67	8.33	13.33
Minimum % error on Validation data	0.00	25.00	10.00	20.00
Confusion Matrix	4 0 0 0	4 0 0 2	6 0 0 0	5 0 0 2
	0 6 0 0	0 3 0 1	0 3 0 1	0 3 0 0
	0 0 4 0	2 0 4 0	0 0 5 0	2 0 4 0
	0 0 0 6	0 0 0 4	0 0 1 4	0 0 0 4

Table 1: Classification summary of MLP on different models of dataset

	Train_1F
No of Hidden Units	50
Learning rate	0.025
Regularization Constant	0.0001
Minimum % error on Training Data	6.67
Minimum % error on Validation data	8.75
Confusion Matrix	20 0 0 0
	0 17 0 3
	0 0 19 1
	0 2 1 17

Table 2: Classification Summary while testing on whole Training set (only for Train_1F model)

Mode	**Color**	**Variable Set**
1B	HS	S5,S2,H7,S6,S9,H8,H11,S12,H1,H12
2B	I	I2,I13,I8,I7,I6,I3
3B	HIS	I2,S5,I10,H11,S1,I13,S13
4B	HIS	All Data Sets

Table 3: Classification models

Model	Color Feature	Greasy Spot	Melanose	Normal	Scab	Overall
1B	HS	100	60	85	60	76.25
2B	I	30	70	50	40	47.5
3B	HIS	100	60	65	50	68.75
4B	All	80	60	40	65	61.25

Table 4: Mahalanobis Distance Classifier

Model	Color Feature	Greasy Spot	Melanose	Normal	Scab	Overall
1B	HS	100	80	85	60	81.25
2B	I	45	65	50	50	52.5
3B	HIS	100	70	60	75	76.25
4B	All	85	65	50	65	66.25

Table 5: Descent Feature Weighted Mahalanobis Distance Classifier

Model	Color Feature	Greasy Spot	Melanose	Normal	Scab	Overall
1B	HS	100	85	100	75	90
2B	I	50	70	60	55	58.75
3B	HIS	100	70	65	80	78.75
4B	All	100	80	100	85	91.25

Table 6: Difference Feature Weighted Mahalanobis Distance Classifier