FarmX: Leaf based disease identification in farms

CS 229 Project Report
Fall 2015

Kushal Chawda **Chanchal Hazra**
{*kchawda,chanchal*}*@stanford.edu*

1. Introduction

The delay and inaccuracy of identification plant diseases is causing significant reduction in both quality and quantity of agricultural products. For example, It is estimated that total losses are amounting to approximately 12% of the produce [1]. Since the current practice of detection and identification of plant diseases is mostly based on visual observation by the experts [2], Automatic detection of plant and related diseases based on a leaf image would be very helpful for the farming world and it will speed up deployment of remedy quickly to reduce or eliminate damage from the disease.

The input to our implementation is a picture of a diseased leaf along with the healthy and diseased portions. The output is the name of the disease that is affecting the leaf. In this project we evaluate several machine learning techniques to (i) Identify the diseased area (We used K-Means and Gaussian Mixture) and (ii) Identify the disease (We used Linear SVM, Quadratic SVM, K-Means and LDA) by classifying among four classes of diseases.

2. Related work

We have seen a few publications where researchers are attempting to automate methods to detect the plant diseases based on images [7] [8] [9].

In [10] used texture CCM features and a Mahalanobis distance based classifier to achieve 95% accuracy. They also used a neural network with lower accuracy numbers. They physically photographed images themselves using a black background.

The Leafsnap system for leaf classification [3], used gaussian mixtures for initial segmentation and used top-hat transformation on top of the segmented image to remove the stem. They generated curvature features using Histogram of curvature over sale and classified images using K-Means.They used the HSV color space

ALRahamneh et al. [7] Used K-means for initial segmentation and used ostu's method for thresholding and masking healthy images. They used mainly texture features and got a high level of accuracy (94%). HSI color space was used here.

In [9], Barbados et al. used intensity histograms from each of different color spaces (HSV, L*a*b* and CMYK) and used modified pairwise voting system to gauge the likelihoods of a particular disease being present in that leaf. Only intensity information was used. They physically photographed the diseased leaves with a black background. They got accuracy ranging from 9% to 100% for selected diseases.

In our view, using a camera independent color space and using texture only features seemed to be good choices to approach the problem. Also, like others we decided to manually collect the data-set, however we used public domain images from the Internet. We also decided to compare the two methods for background separation (k-means vs Gaussian Mixtures). Unlike others, we limited the number of texture features to just Contrast, Correlation, Energy and Homogeneity.

3. The Dataset

Since the dataset for diseased leaf images were not publicly available, we used a subset of images having Creative Commons Licence from the flickr website of Dr. Scott Nelson [4]. We manually downloaded images for four categories of diseases:

- Bacterial Blight(26 images)
- Rust(36 Images)
- Corynespora Leaf Spot(43 Images)
- Mildew (26 Images)

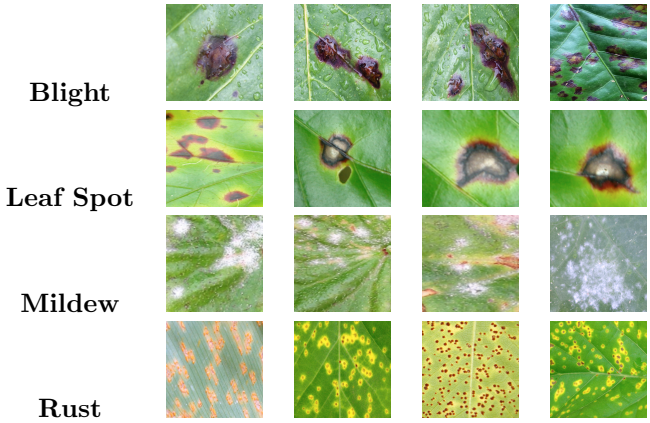

Fig. 1. Sample Images from our data-set

4. High Level Framework

The high level framework is outlined as follows:

- Background Removal: We wanted to extract features from only the diseased portion of the leaf.

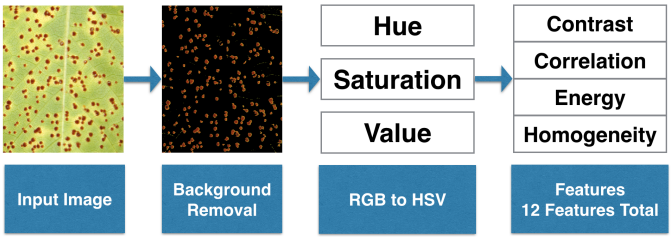

Fig. 2. Image Processing Pipeline

Hence this step removes the healthy portion of the leaf.

- Feature Extraction: We chose only texture features (Contrast, Correlation, Energy and Homogenity), since we found our dataset contained diseases of varying texture and wanted to extract only the texture features.
- Classification and Error analysis: We consider four models (Linear SVM, Quadratic SVM, LDA and K-Means) for classification and generate confusion matrices and performance metrics.

5. Background Removal

A. *Choosing the color space*

Since our dataset had images with varying lighting conditions, we wanted to choose a color space that is more resilient to this [5]. We chose the (Hue, Saturation, Value) HSV color spaces since the hue value remains same for varying lighting conditions. We also found a convenient function in Matlab(rgb2hsv) to convert RGB to HSV, which made it a good choice to start with. Fig. 3 shows a comparison between RGB and HSV channels of the input image.

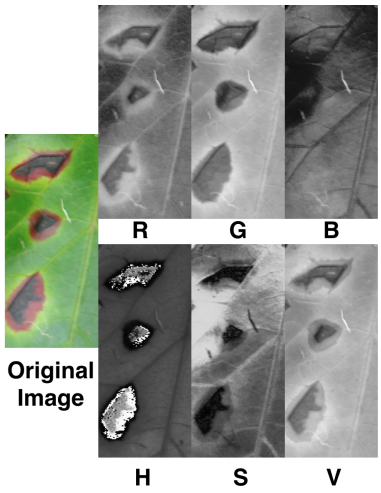

Fig. 3. RGB vs HSV Color Spaces

B. *Image segmentation*

Since our dataset contained images having varying lightning conditions, we wanted to choose a color space that is more resilient to changes in lighting conditions [5]. We chose the (Hue, Saturation, Value) HSV color spaces since the hue value remains same for varying lighting conditions. We used rgb2hsv function in Matlab to do this conversion. Fig. 3 shows the various channels of a diseased leaf, comparing RGB channels with HSV channels. In this figure it can be seen that the S and V are dominant for the diseased part and were thus used for segmentation.

1. *k-means vs Mixture of Gaussians*

We wanted to compare k-means vs. mixture of Gaussians for segmentation and also wanted to arrive at optimal value of the number of segments(k). We found good visual results with k=4 for k-means and 4 Gaussians for Gaussian mixture. The reasoning behind selecting four was that the leaf would be segmented by (i) healthy areas, (2) diseased areas, (3)leaf stem, (4)camera reflection or water).

For each of the identified segments, we created four images which masked the pixels that did not belong to the identified segment (All three components set to zero). The advantage of doing this was that the segmented portions could be viewed in the RGB space for visual inspection (without conversion). Also since the segmentation did not use the Hue channel for k-means or EM, it can now be potentially used by converting back from the identified cluster segments in the RGB space. Results of this process is shown in Figure 4. Here we can see that in case of EM with four Gaussians, the diseased part includes small part of the stem and surrounding healthy areas, while in case of k-means separation is more distinct. However, in case of k-means the area of the diseased part is lower due to the hard separation. This was actually preferred, because our feature is texture based and we don't want to include healthy portions of the leaf with the diseased part.

2. *Choosing the diseased segment*

In order to select the segment containing the diseased image, we average the hue portion of the non zeroed pixels across each segmented image and select the image that is farthest away from the hue value of green (0.3333 in Matlab). While this method works on most images, this might not work for leaves where the healthy color is not green. In such cases we manually correct set diseased image to correct segment. Since the number of such images were limited, we were easily able to do this. This was also an important step since we did not want to incorrectly use healthy images to training disease classifiers.

6. Features

A. *Feature Extraction*

We used Gray-Level Co-occurrence Matrix (GLCM) based features on each HSV channel of the segmented image containing diseased areas. A GLCM matrix shows how often a pixel with the intensity (gray-level) value i

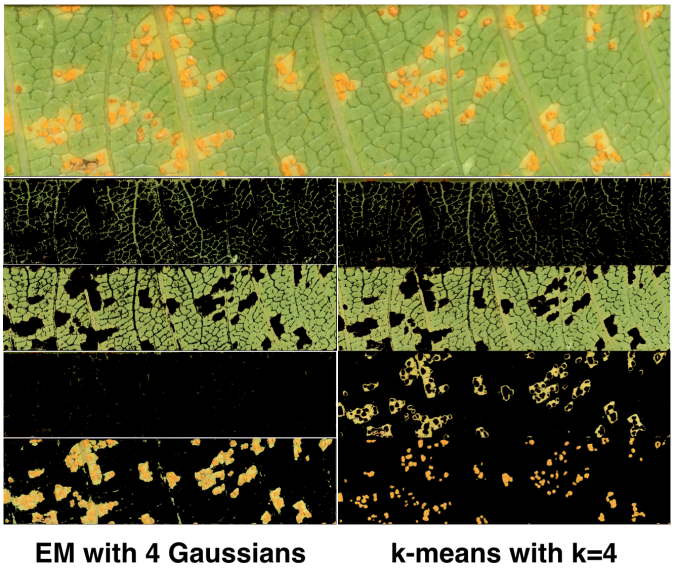

Fig. 4. Segmentation using EM vs k-means (4 parts)

occurs in a specific spatial relationship to a pixel with the value j. Features were thus defined as follows:

- **Contrast**: Measures local variations in the gray-level co-occurrence matrix.

$$\left(\sum_{i,j}|i-j|^2 p(i,j)\right)$$

- **Correlation**: Measures joint probability occurrence.

$$\left(\sum_{i,j}\frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i\sigma_j}\right)$$

- **Energy** Provides the sum of squared elements in the GLCM.

$$\left(\sum_{i,j}p(i,j)^2\right)$$

- **Homogeneity**: Measures closeness of the distribution of elements in the GLCM to the GLCM diagonal.

$$\left(\sum_{i,j}\frac{p(i,j)}{1+|i-j|}\right)$$

We extracted these four features from each of Hue, Saturation and Value channel, resulting in 12 featurs in total. We wanted to first start with these features to see how the results were before deciding to modify them. We used Matlab's 'graycomatrix' and 'graycoprops' functions to extract the features from the selected HSV segmented image channels.

B. Feature Selection

Given these 4 features(Contrast, Correlation, Energy and Homogeneity in order) from the Hue, Saturation and Value channels each respectively, we wanted to select the subset of these 12 features (F1-F12) that minimize the cross validation error. We tried multiple models and found the quadratic-SVM gave us best results. As seen from Fig. 5, We chose features [F5,F6,F9,F10,F11] since this gave us the best results and also because it completely eliminated the Hue channel from feature selection. Eliminating the Hue channel is desirable since it makes our features more resilient to varying color conditions.

Feature Combination	Accuracy (5 Fold CV) (Quadratic SVM)
[5,6,9,10,11]	93.89%
[3,5,9,10,12]	92.37%
[4,6,8,9,10]	91.60%
[4,6,9,10]	90.84%
[5,8,9,10,11]	90.08%
[5,9,10,11]	89.31%
[8,9,10,11]	88.55%
[1,8,9]	87.79%
[6,7,9,10]	86.26%

Fig. 5. Top ten feature combination using forward search

Features 5 and 9 seemed to be dominant in being able to visually cluster the subset of our selected feature set. As seen in Fig. 6, we are able to visually cluster the data after making a scatter plot of F5, vs F9

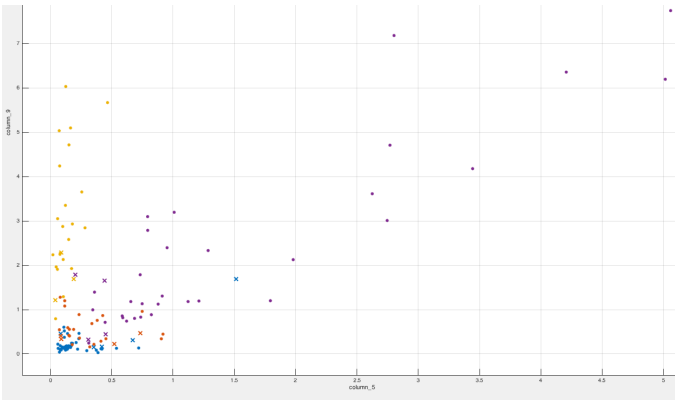

Fig. 6. Scatter plot of F5 vs F9 (Contrast in S channel vs Contrast in the V Channel

7. Models

We used SVM (Linear and Quadratic), K-Means and LDA to train and compare classification performance:

SVM: An SVM classifies data that has exactly two classes by finding the best hyperplane that separates all

points of one class from the other class. The best margin hyperplane that maximizes the margin between the two classes is given by the following optimization problem:

$$min_{\gamma,w,b}\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\epsilon_i$$

$$s.t.\ y^{(i)}(w^T x^{(i)} + h) \geq 1 - \epsilon_i, i = 1, ..., m$$

$$\epsilon_i \geq 0, i = 1, ..., m$$

For SVM, we used an **L1 soft margin classifier with C=1**.

In order to achieve multi-class classification, we use the **One-vs-One** approach which constructs one classifier per pair of classes. During prediction, the class which received the most votes is selected. In the event of a tie the class with the highest aggregate classification confidence is selected. This method thus builds $N(N-1)/2$ classifiers, where N is the number of classes.

We compared results across two kernel functions:

- **Linear Kernel**: The linear kernel has the form:

$$K(x_i, x_i) = \sum_{j=1}^{p} x_{ij}x'_{ij}$$

- **Quadratic Kernel**: Is a polynomial kernel and has a much more flexible decision boundary:

$$K(x_i, x_i) = (1 + \sum_{j=1}^{p} x_{ij}x'_{ij})^2$$

We also wanted to use discriminative models:

- **LDA**: We used mixtures of Gaussians with a diagonal covariance matrix to model the parameters of a Gaussian for each class. The prediction tries to minimize the classification cost given by:

$$y = argmin_{y=1...K}\sum_{k=1} KP(k|x)C(y|k)$$

  Where, y is the predicted classification, K is the number of classes, $P(k|x)$ is the posterior probability of class k for observation x and $C(y|k)$
- **k-means**:We use K-means as a classification and also as a segmentation algorithm. For segmentation, the points are labelled belonging to k=4 clusters, and for classification the point is classified based on the closest cluster centroid.

8. Results

We used several classification models with **5-fold** cross validation to compare learning algorithms:

We used Matlab to compare results across: **k-means, Linear Discriminative Analysis, Linear SVM and Quadratic SVM**. The optimized feature selection gave us a 2% improvement over the previous set of results (with all 12 features). All the models were trained and compared with the optimized feature selection [F5,F6,F9,F10,F11].

Models	Accuracy (5 Fold CV)	
	k-means (k=4) B/G Removal	EM with 4 Gaussians B/G Removal
Quadratic SVM	93.1%	84.8%
Linear SVM	90.1%	84.2%
KNN	87.8%	84.2%
LDA	73.3%	67.7%

Fig. 7. Overview of results comparing background segmentation methods (EM and k-means) along with the performance for various learning algorithms

Fig. 7 shows us the overall accuracy numbers. We see accuracy of 93.1% on Quadratic SVM with K=4 background removal. This makes sense for a texture based feature set because we need the hard separation. Confusion Matrices are shown in Fig 8 and 10. We see in case of the Linear SVM the maximum miss-classification seems to be between Blight and Leaf Spot, which also makes sense because they are visually similar as seen in Fig. 1

We see that LDA does not perform as well as other models. In case of Gaussian mixture based segmentation, We see that the texture based classification seems sensitive to "noise" in the texture based features. We were able to visually confirm (As seen in Fig.4) that diseased portion of the images segmented using GM included healthy areas as well.

9. Conclusion/Future Work

For the best performing model (Quadratic SVM with K-means background separation with K=4), chosen with optimized feature set, we see our accuracy is 93.1%. The Precision, Recall and F1 scores for each classifier are shown in Fig. 9 are within 85% to 100%.

Since we have fairly high accuracy, our next step would be to focus on recommendations to treat the disease. In order to do this better, we would have to identify the species and build a dataset having recommendations.

As seen from the EM case, our model is sensitive to inaccuracies in background removal, we would thus focus on making it robust to be able better select the diseased cluster and ensuring the diseased cluster does not have healthy areas. Given more time, we could train a classifier to just recognize diseased segments of the image to make the background separation more robust.

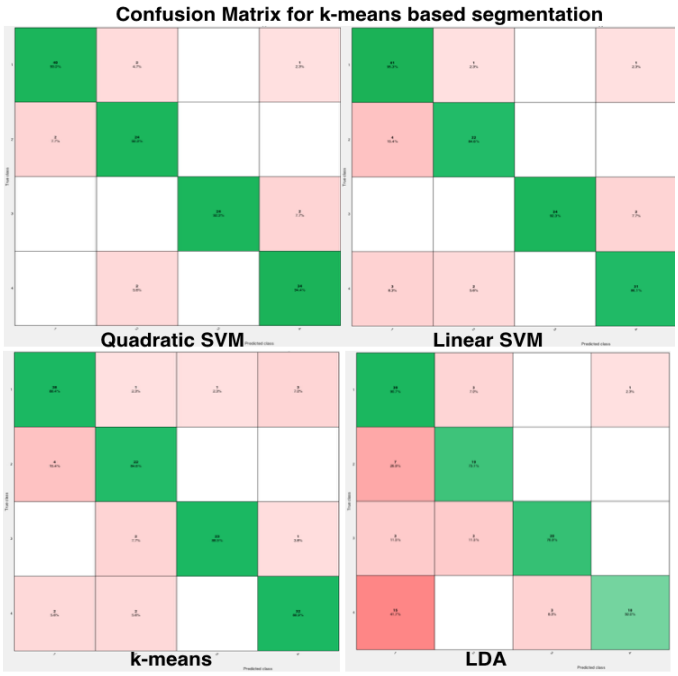

Fig. 8. Confusion Matrix with k-means B/G separation for (1: Leaf Spot), (2: Blight), (3: Mildew) and (4: Rust)

Algo/Disease		EM			K-Means		
		Precision	Recall	F1	Precision	Recall	F1
LDA	1	50.8%	91.4%	0.65	60.9%	90.7%	0.73
	2	70.6%	44.4%	0.55	76.0%	73.1%	0.75
	3	78.9%	68.2%	0.73	87.0%	76.9%	0.82
	4	81.4%	64.9%	0.72	94.7%	50.0%	0.65
Q-SVM	1	83.8%	88.6%	0.86	95.2%	93.0%	0.94
	2	63.6%	77.8%	0.70	85.7%	92.3%	0.89
	3	87.5%	63.6%	0.74	100.0%	92.3%	0.96
	4	94.4%	91.9%	0.93	91.9%	94.4%	0.93
L-SVM	1	81.1%	85.7%	0.83	85.4%	95.3%	0.90
	2	75.9%	81.5%	0.79	88.0%	84.6%	0.86
	3	76.2%	72.7%	0.74	100.0%	92.3%	0.96
	4	91.5%	87.8%	0.90	91.2%	86.1%	0.89
K-Means	1	83.3%	85.7%	0.85	86.4%	88.4%	0.87
	2	79.2%	70.4%	0.75	81.5%	84.6%	0.83
	3	93.3%	63.6%	0.76	95.8%	88.5%	0.92
	4	84.3%	94.6%	0.89	88.9%	88.9%	0.89

Fig. 9. Performance Results with k-means B/G separation for (1: Leaf Spot), (2: Blight), (3: Mildew) and (4: Rust)

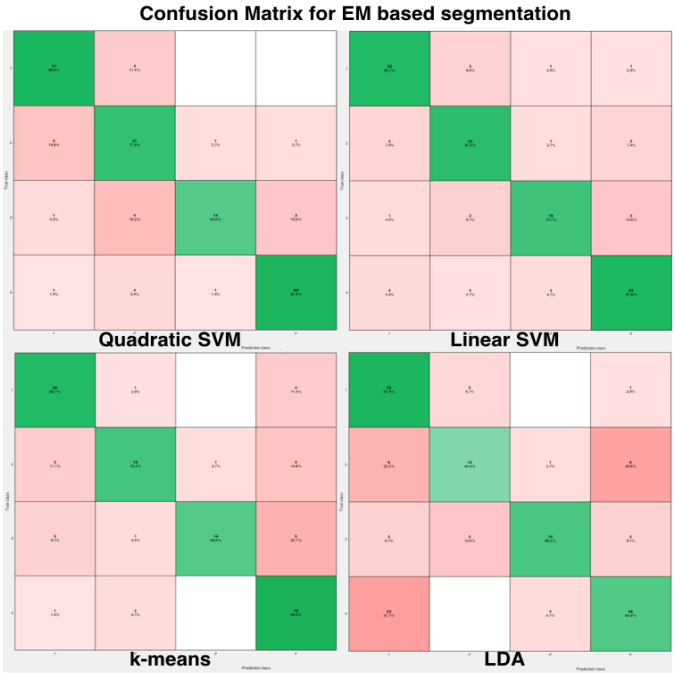

Fig. 10. Confusion Matrix with Gaussian Mixture B/G separation for (1: Leaf Spot), (2: Blight), (3: Mildew) and (4: Rust)

References

1. Martinez-Espinoza Alfredo, et al. "GEORGIA PLANT DISEASE LOSS ESTIMATES" *UGA Extension AP 102-6* 2013
2. Sastry, K. Subramanya, A. Zitter, Thomas. "Management of Virus and Viroid Diseases of Crops in the Tropics" *Springer* (2014).
3. Neeraj Kumar and Peter N. Belhumeur and Arijit Biswas and David W. Jacobs and W. John Kress and Ida Lopez and Joo V. B. Soares "Leafsnap: A Computer Vision System for Automatic Plant Species Identification," *http://leafsnap.com/dataset/* (2012).
4. *https://www.flickr.com/photos/scotnelson* .
5. HSV Color Space: *https://en.wikipedia.org/wiki/HSL_and_HSV*.
6. Forcyth, Ponce. "Computer Vision, a modern approach" ISBN:0130851981 (2002)
7. H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh "Fast and Accurate Detection and Classification of Plant Diseases" International Journal of Computer Applications (0975 8887) Volume 17 No.1, March 2011
8. Shi Yun1, Wang Xianfeng1, Zhang Shanwen1, Zhang Chuanlei "PNN based crop disease recognition with leaf image features and meteorological data" Int J Agric Biol Eng, Aug 2015
9. Jayme Garcia Arnal Barbedo1, Cludia Vieira Godoy "Automatic Classification of Soybean Diseases Based on Digital Images of Leaf Symptoms " SBI AGRO (Oct 2015)
10. Pydipati, R., T. F. Burks, and W. S. Lee. "Statistical and neural network classifiers for citrus disease detection using machine vision." TRANSACTIONS-AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS 48.5 (2005): 2007.