Disease Detection in Leaves

A Project Report

Submitted in fulfilment of the

Requirement for the award of the degree of

Master of Computer Applications

(Session 2016 - 2017)

To

By

Sushmita Sharma (5141079)

Harshit Bhatia (5141053)

Devendra Pratap Singh (5141055)

Under the Supervision

of

Dr. Kapil

Dept. of Computer Applications

DECLARATION

We hereby declare that the work which is being presented in this project report entitled **“**Disease Detection in Leaves” requirement for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** submitted to Department of Computer Applications, National Institute of Technology, Kurukshetra is an authentic work done by us during a period from August 2015 to November 2016 under the guidance of Dr. Kapil**.**

The work presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Sushmita Sharma	Harshit Bhatia	Devendra Pratap Singh
(5141079)	(5141053)	(5141055)

This is to certify that the above statement made by the candidate is correct to best of my knowledge and belief.

Date :

Place : Dr. Kapil

ACKNOWLEDGEMENT

First and foremost, we express our heartfelt gratitude towards **Prof. Ashutosh Kumar Singh**, HoD, Department of Computer Applications, National Institute of Technology, Kurukshetra and **Dr. Kapil**, Department of Computer Applications, National Institute of Technology, Kurukshetra for attending to all our queries and providing constant guidance to us despite of their busy schedule.

We also thank Prof. Sarika Jain, Mr. Prantik Biswas and all the other faculty members for their constant guidance and motivation.

A special word of thanks to our friends who have always heard our ideas and have given us positive feedback.

ABSTRACT

According to studies, relying on naked eye observation of plants is not accurate. Providing automatic and accurate image-processing-based solution for this task can be of great realistic significance.

In this report, firstly, we have discussed the various algorithms that can be implemented to aid the process of automatic leaf recognition` by the computer. The algorithms that stand as options are Probabilistic Neural Network (PNN), Support Vector Machine (SVM) and K-Means Algorithm. A comparative analysis of the various algorithms has also been presented to evaluate the algorithms on the basis of accuracy.

We have also discussed the different techniques that can be implemented for plant leaf disease classification. Classification is a technique where a leaf is classified based on its different morphological features. The classification opted in this report is k-means. Plant leaf detection and disease classification has wide applications in various fields such as in biological research in agriculture.

LIST OF FIGURES

FIGURE NO.	FIGURE CAPTION	PAGE NO.
Figure 1	K-means clustering	7
Figure 2	PCA implementation	10
Figure 3	PNN implementation	11
Figure 4	Screenshot for output window	14
Figure 5	Screenshot for output window	14
Figure 6	Screenshot for output window	15
Figure 7	Screenshot for output window	15
Figure 8	Screenshot for output window	16
Figure 9	Screenshot for output window	16
Figure 10	Screenshot for output window	17
Figure 11	Screenshot for output window	17

LIST OF TABLES

TABLE NO.	TABLE CAPTION	PAGE NO.
Table 1	Table illustrating the literature survey	2

TABLE OF CONTENTS

Declaration		i
Acknowledgement		ii
Abstract		iii
List of Figures		iv
List of Tables		v
Table of contents		vi

Chapter No.	**Description**	**Page No.**
Chapter 1	Introduction	1
Chapter 2	Literature survey	2
	2.1 Literature survey / Previous work	3
	2.2 Research gaps	4
Chapter 3	Project objectives	5
Chapter 4	Theoretical modelling	
	4.1 Proposed concept model or algorithm	6
	4.2 Technical specifications	13
Chapter 5	Implementation, Simulation and Results	
	4.1 Snapshots of execution of the code	14
	4.2 Installation instructions	18
Chapter 6	Conclusion and future scope	19
	References	20

Chapter 1
INTRODUCTION

Plants play an important role in both human life and other lives that exist on the earth. Due to environmental deterioration and lack of awareness, many rare plant species are at the margins of extinction. Leaf and disease classification and recognition plays a vital role in all these endeavours.

Today, thousands of plants are on the verge of extinction and are still unknown to the human world. The inclusion of such bio-diversity to the data set will certainly be a major achievement. In most of the countries farmers detect pests manually through their observation of naked eyes, which requires continuous monitoring of the crop stems and leaves, which is a difficult, labor intensive, inaccurate and expensive task for large farms [6].

It is also a long discussed topic on how to extract or measure leaf features [7], [10]. That makes the application of pattern recognition in this field a new challenge [11] [9]. According to [11], data acquisition from living plant automatically by the computer has not been implemented.

We have discussed the various algorithms that can be implemented to aid the process of automatic leaf recognition` by the computer. The algorithms that stand as options are Probabilistic Neural Network (PNN), Support Vector Machine (SVM) and K-Means Algorithm. A comparative analysis of the various algorithms has also been presented to evaluate the algorithms on the basis of accuracy.

For the purpose of disease detection, we have implemented k-means algorithm. The identification of leaf is the primary step towards leaf classification. Once a leaf has been identified, the scope of disease detection narrows to a particular leaf type, hence making the system more accurate.

Chapter 2

LITERATURE SURVEY

Table 1: **Table illustrating the literature survey**

S.no	Author Name	Name of the paper	Conclusion	Limitations
1.	Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang and Qiao-Liang Xiang	A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network	1. Addresses the risk of plant extinction and present how to teach a computer how to classify plants. 2. Presents how to process an image of a leaf before any processing of image, to extract the features of that image. 3. Presents the working of PCA (Principal Component Analysis) and Introduction of PNN (Probabilistic Neural Network).	Algorithm is generating only index of greatest value not the second greatest value.
2.	H. Al-Hiary, S. Bani-Ahmad, M.Reyalat, M. Braik and Z. ALRahamneh	Fast and Accurate Detection and Classification of Plant Diseases.	1. Propose and experimentally evaluate a software solution for automatic detection and classification of plant leaf diseases. 2. Presents the working of K-means clustering algorithm and SGDM matrix.	1. Difficult to predict the K-value. 2. It does not work well with different cluster size.

S.no	Author Name	Name of the paper	Conclusion	Limitations
3.	Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang	Extreme Learning Machine for Regression and Multiclass Classification.	1. Proposed a concept of Extreme learning machine (ELM), that provides a unified learning platform with a widespread type of feature mappings. 2. Further proposed PSVM (Proximal support vector machine).	1. Randomness of ELM causes an additional uncertainty problem, both in approximation and learning.
4.	Aakanksha Rastogi, Ritika Arora, Shanu Sharma	Leaf Disease Detection and Grading using Computer Vision Technology &Fuzzy Logic	1. Presents the basics of computer vision, colour image segmentation. 2. Presents the concept of fuzzy logic and Artificial Neural Network (ANN).	1. Difficult to work with images of leaves that can not be used for feature extraction. 2. Damaged leaf 3. Size/Shape. 4. Leaf angles.
5.	Dheeb Al-Bashish, Malik Braik and Sulieman Bani-Ahmad.	Detection and Classifications of leaf Diseases using K-means based segmentation and Neural Network based Classification	1. Presents the idea of K-means clustering in disease detection. 2. Present How Texture feature analysis and Hue Saturation analysis works	1. Difficult to decide K-value for K-means clustering

RESEARCH GAP

According to the literature survey we have an idea of how this system works for one leaf. We have extended our project to work on 32 kinds of leaves with a large dataset of 1800 images of these leaves.

While extracting the features of defected leaf we are working on boundary value and we are using values of Hue-saturation analysis and texture analysis for detection of leaf disease.

Chapter 3

PROJECT OBJECTIVES

With the continuous increase in leaf diseases, it is not possible for humans to detect each with the naked eye, neither it is feasible. The cost involved is too high, since the crops are at stake. There are large data sets available online which help us to identify multiple leaves and diseases.

The biggest limitation of existing systems is that the diseases of a only particular leaf are detected by the system. We have tried to implement a system that is able to identify the leaf and then the disease it is infected with, hence enhancing the scope. The comparison between various available algorithms has helped us to identify the best accuracy giving algorithm on the basis of closeness to actual result. This should prove a milestone in the identification of diseases, which can help to curb and control some lethal diseases in leaves.

Chapter 4

THEORETICAL MODELLING

- **Proposed concept and algorithm**
- **K-means Algorithm for Clustering the Disease Part in Leaf**

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center). In other words, its objective is to find:

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community.

Given an initial set of k means m1(1),…,mk(1) (see below), the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean.

$$S_i^{(t)} = \big\{ x_p : \big\| x_p - m_i^{(t)} \big\|^2 \le \big\| x_p - m_j^{(t)} \big\|^2 \ \forall j, 1 \le j \le k$$

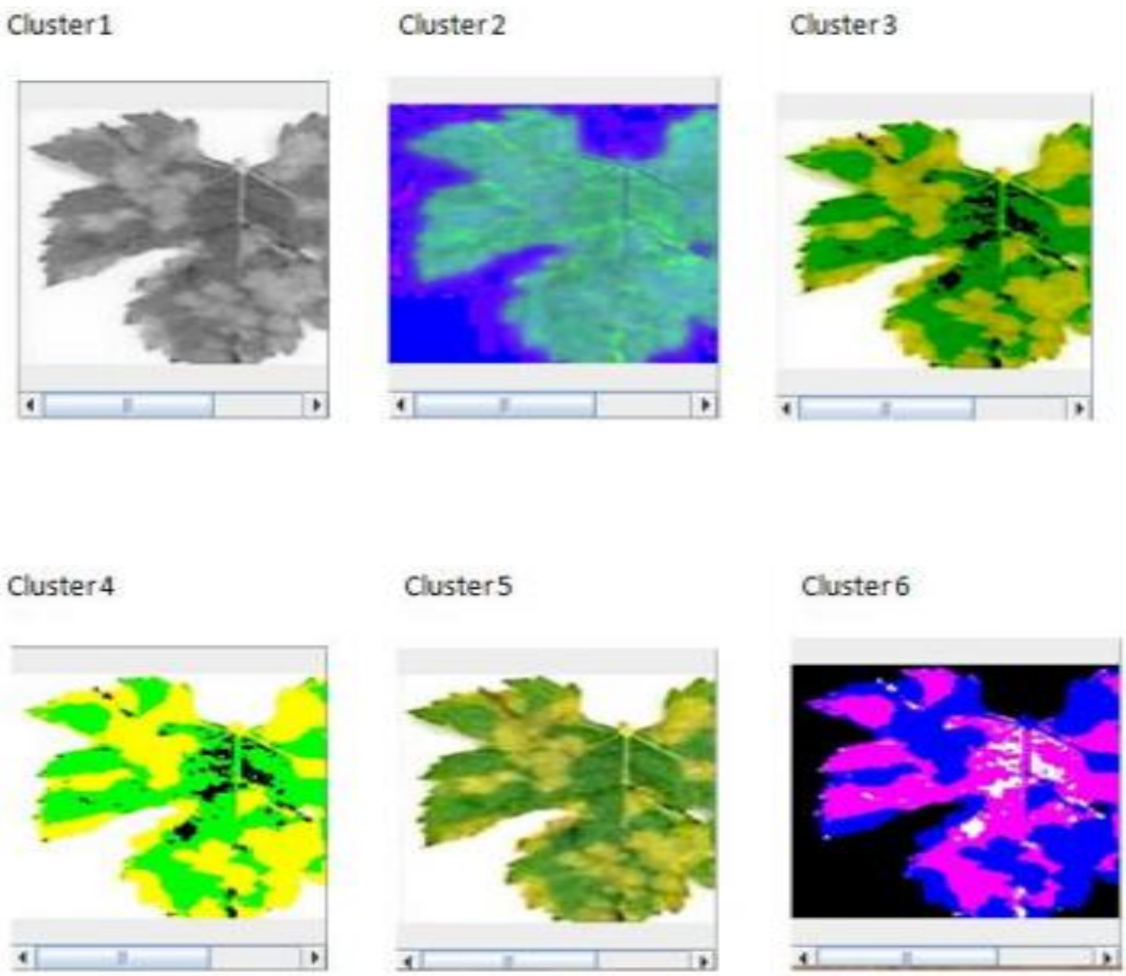
$$S_i^{(t)} = \big\{ x_p : \big\| x_p - m_i^{(t)} \big\|^2 \le \big\| x_p - m_j^{(t)} \big\|^2 \ \forall j, 1 \le j \le k \big\},$$

where each Xp is assigned to exactly one S^(t) even if it could be assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective.



Six cluster formed by K-Means clustering

Fig. 1

- **PCA (Principle Component Analysis) Algorithm for Feature Extraction and Minimize the Input Vector Size for PNN**

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Consider a data matrix, X, with column-wise zero empirical mean (the sample mean of each column has been shifted to zero), where each of the n rows represents a different repetition of the experiment, and each of the p columns gives a particular kind of feature.

Mathematically, the transformation is defined by a set of p-dimensional vectors of weights or loadings Wk=(w1,….Wp)(k) that map each row vector X(i) of X to a new vector of principal component scores t(i)=(t1,….,tk)(i), given by

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$$

In such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from x, with each loading vector w constrained to be a unit vector.

First component

The first loading vector w(1) thus has to satisfy

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

Equivalently, writing this in matrix form gives

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \{ \|\mathbf{X}\mathbf{w}\|^2 \} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \right\}$$

Since $\mathbf{w}_{(1)}$ has been defined to be a unit vector, it equivalently also satisfies

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

The quantity to be maximised can be recognised as a Rayleigh quotient. A standard result for a symmetric matrix such as X^(t)X is that the quotient's maximum possible value is the largest eigenvalue of the matrix, which occurs when w is the corresponding eigenvector.

With w(1) found, the first component of a data vector x(i) can then be given as a score t1(i) = x(i) · w(1) in the transformed co-ordinates, or as the corresponding vector in the original variables, {x(i) · w(1)} w(1).

Further components

The kth component can be found by subtracting the first k − 1 principal components from X:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X} \mathbf{w}_{(s)} \mathbf{w}_{(s)}^{\mathrm{T}}$$

And then finding the loading vector which extracts the maximum variance from this new data matrix

$$\mathbf{w}_{(k)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg\max \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

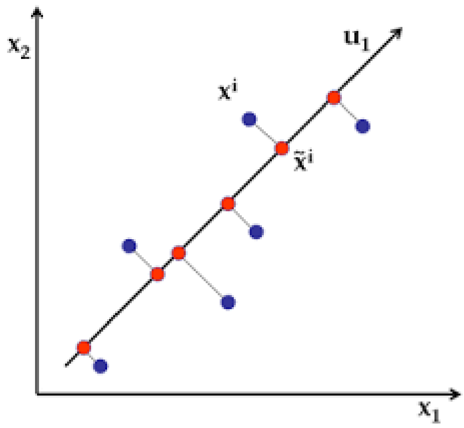
It turns out that this gives the remaining eigenvectors of X^(T)X, with the maximum values for the quantity in brackets given by their corresponding eigenvalues. Thus the loading vectors are eigenvectors of X^(T)X.

The kth component of a data vector x(i) can therefore be given as a score tk(i) = x(i) · w(k) in the transformed co-ordinates, or as the corresponding vector in the space of the original variables, {x(i) · w(k)} w(k), where w(k) is the kth eigenvector of X^(T)X.

The full principal components decomposition of X can therefore be given as

$$\mathbf{T} = \mathbf{X}\mathbf{W}$$

Where W is a p-by-p matrix whose columns are the eigenvectors of X^(T)X



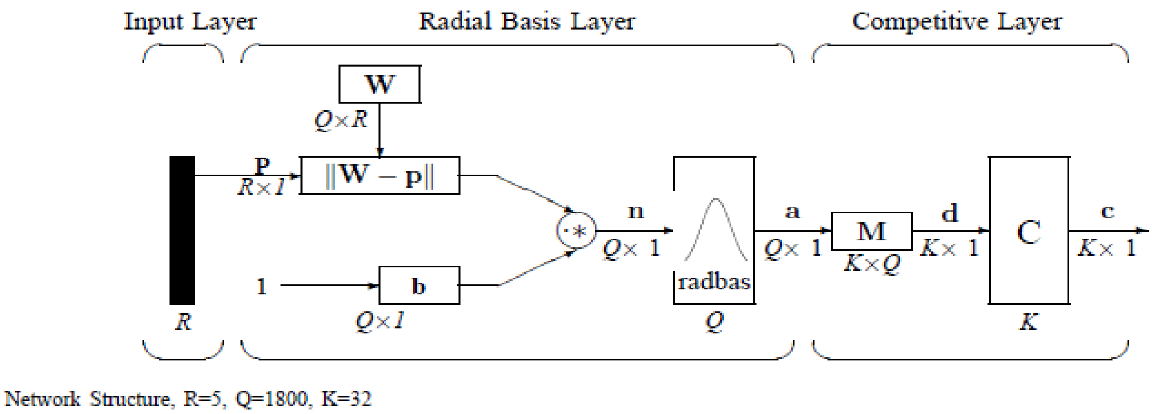
PCA Diagram
Fig-2

- PNN

An artificial neural network (ANN) is an interconnected group of artificial neurons simulating the thinking process of human brain. One can consider an ANN as a "magical" black box trained to achieve expected intelligent process, against the input and output information stream. Thus, there is no need for a specified algorithm on how to identify different plants. PNN is derived from Radial Basis Function (RBF) Network which is an ANN using RBF. RBF is a bell shape function that scales the variable nonlinearly.

A probabilistic neural network (PNN) is a feed forward neural network, which was derived from the Bayesian network and a statistical algorithm called Kernel Fisher discriminant analysis. It was introduced by D.F. Specht in the early 1990s. In a PNN, the operations are organized into a multi-layered feed forward network with four layers:

The network classifies input vector into a specific class because that class has the maximum probability to be correct. In this paper, the PNN has three layers: the Input layer, Radial Basis Layer and the Competitive Layer. Radial Basis Layer evaluates vector distances between input vector and row weight vectors in weight matrix. These distances are scaled by Radial Basis Function nonlinearly. Then the Competitive Layer finds the shortest distance among them, and thus finds the training pattern closest to the input pattern based on their distance.



PNN Working Layers

Fig-3

Network Structure:

The network structure in our purposed scheme is illustrated in Fig. 3. We adopt symbols and notations used in the book Neural Network Design. These symbols and notations are also used by MATLAB Neural Network Toolbox. Dimensions of arrays are marked under their names.

1) **Input Layer**: The input vector, denoted as p, is presented as the black vertical bar in Fig. 5. Its dimension is R × 1. In this project, R = 5.

2) Radial Basis Layer: In Radial Basis Layer, the vector distances between input vector p and the weight vector made of each row of weight matrix W are calculated. Here, the vector distance is defined as the dot product between two vectors [29]. Assume the dimension of W is Q×R. The dot product between p and the i-th row of W produces the i-th element of the distance vector ||W−p||, whose dimension is Q×1, as shown in Fig. 3. The minus symbol, "–", indicates that it is the distance between vectors.

Then, the bias vector b is combined with ||W− p|| by an element-by-element multiplication, represented as "· ∗" in Fig. 3. The result is denoted as n = ||W− p|| · ∗p. The transfer function in PNN has built into a distance criterion with respect to a center. In this paper, we define it as

$$radbas(n) = e^{-n^2}$$

Each element of n is substituted into upper equation and produces corresponding element of a, the output vector of Radial Basis Layer. We can represent the i-th element of a as

$$a_i = radbas(||\mathbf{W}_i - \mathbf{p}|| \cdot *\mathbf{b}_i)$$

Where Wi is the vector made of the i-th row of W and bi is the i-th element of bias vector b.

TECHNICAL SPECIFICATIONS

Language used: MATLAB Script

- Matlab script code tends to be more readable and compact. Since the code for Pre-Processing the image involves a lot of complexity thus we can use direct function in MATLAB Tool Box.
- The chances for making syntax errors reduce considerably because of the indentation rules Matlab Script enforces.
- One doesn’t need to declare the variables before using them. Matlab is a strongly typed language.
- For Matlab we can use Octave, Octave is open source so one can always look how things have been implemented.
- Matlab Tool Box offers a wider set of choices in Mathematical packages and toolsets.

Chapter 5

IMPLEMENTATION AND RESULTS

1. Introduction of Software

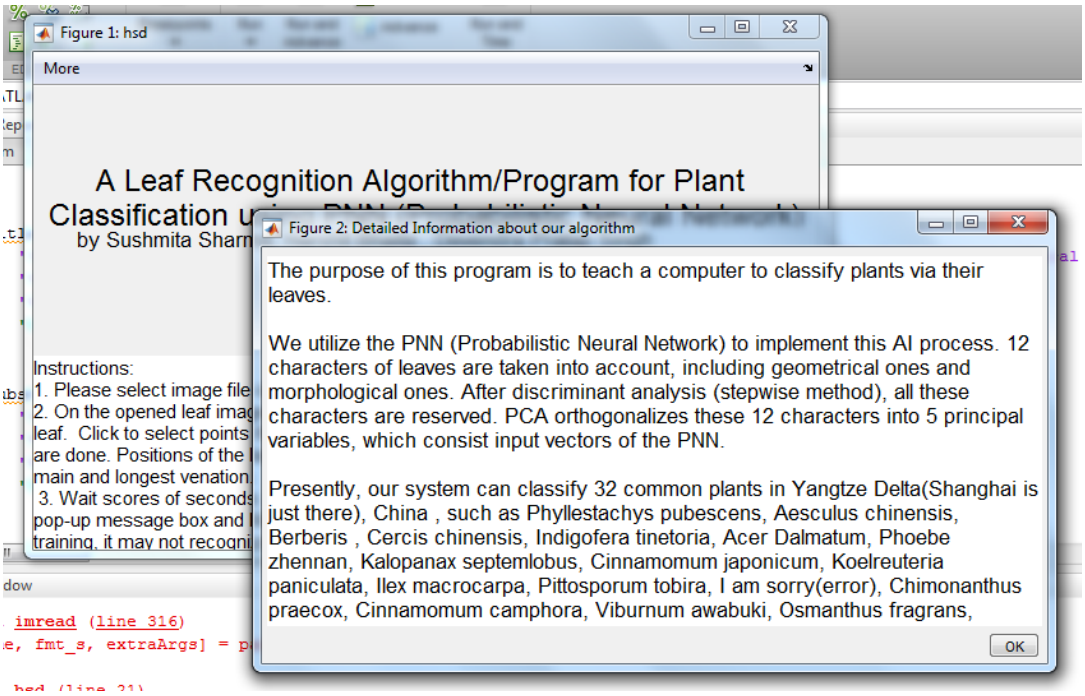


Fig-4

2. Input Leaf for Pre-Processing

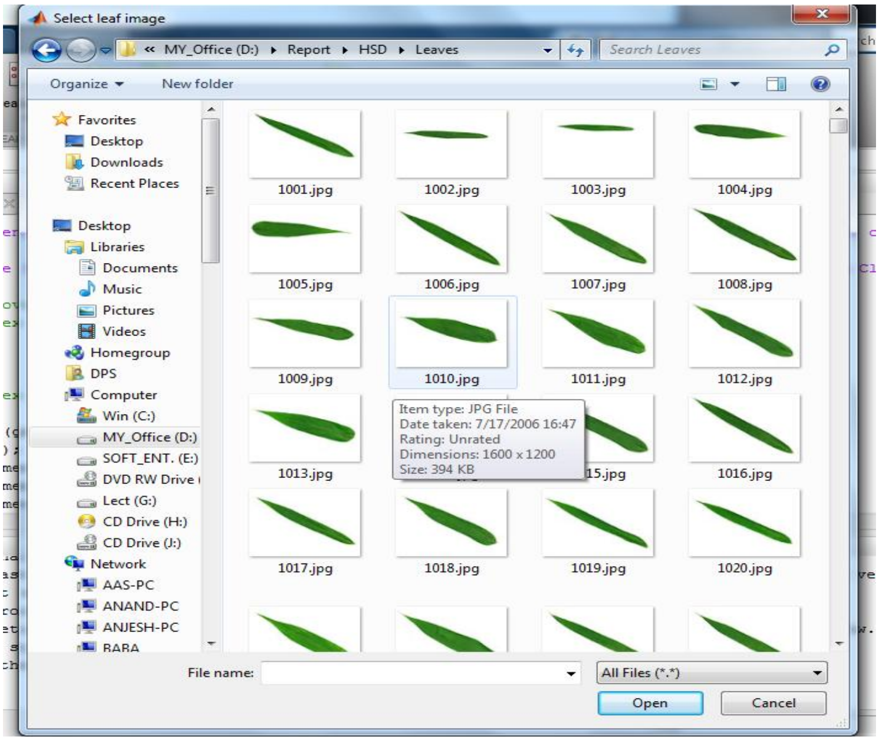


Fig -5

3. Select the leaf terminals

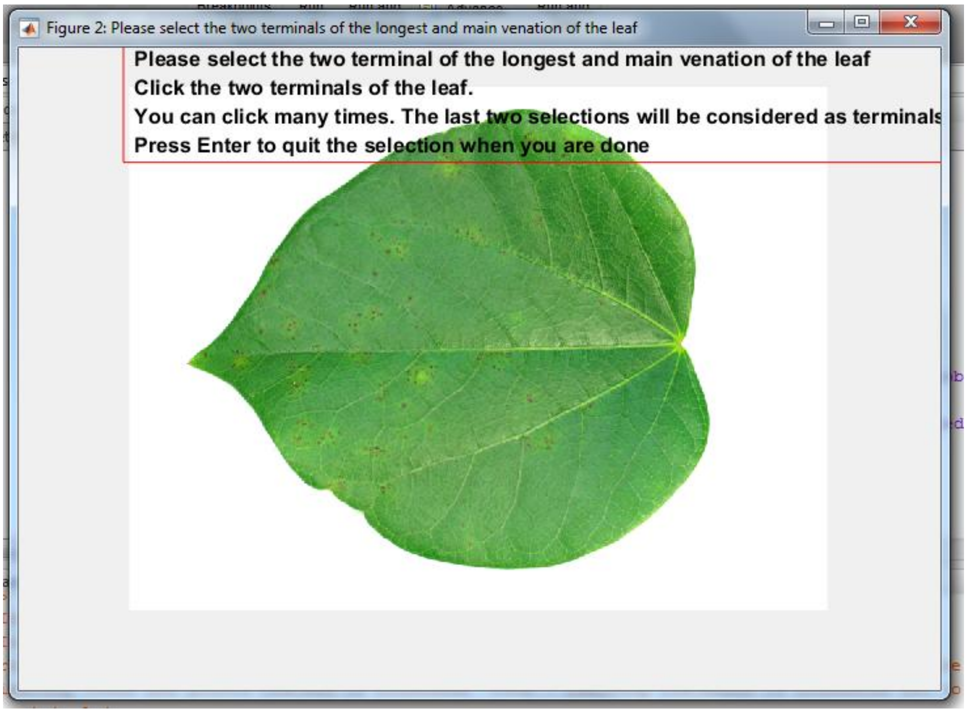


Fig-6

4. Result for Input Leaf

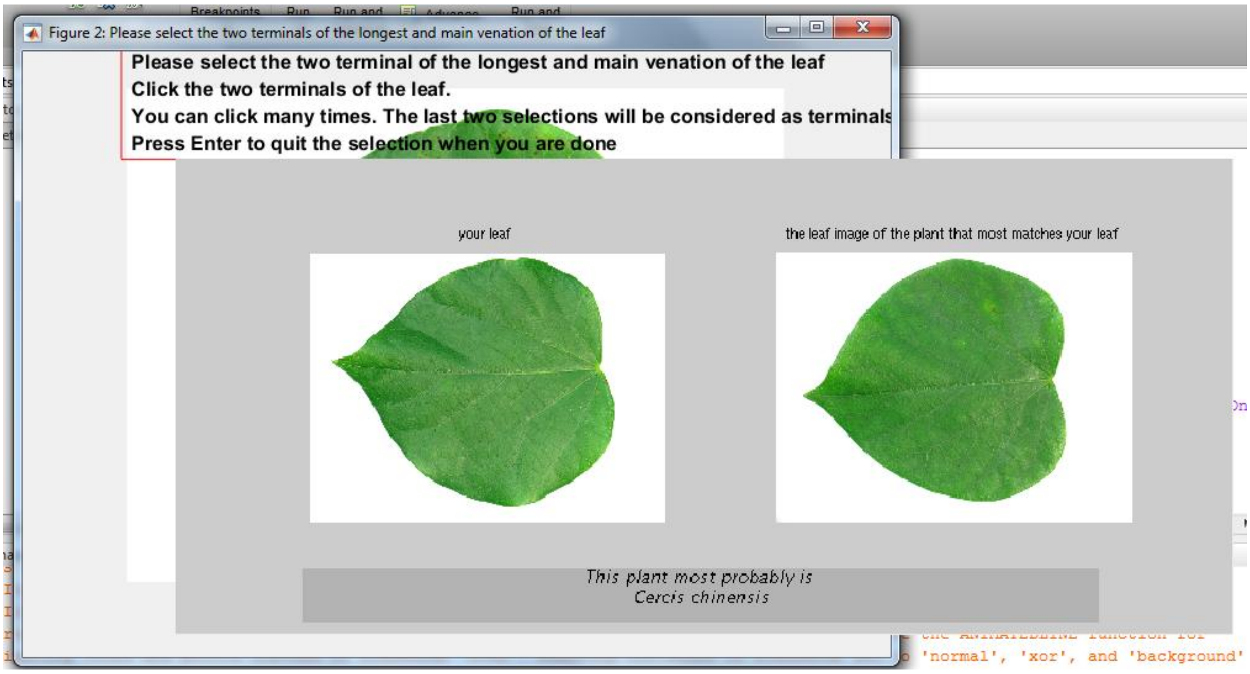


Fig-7

5. Input of Defected Leaf

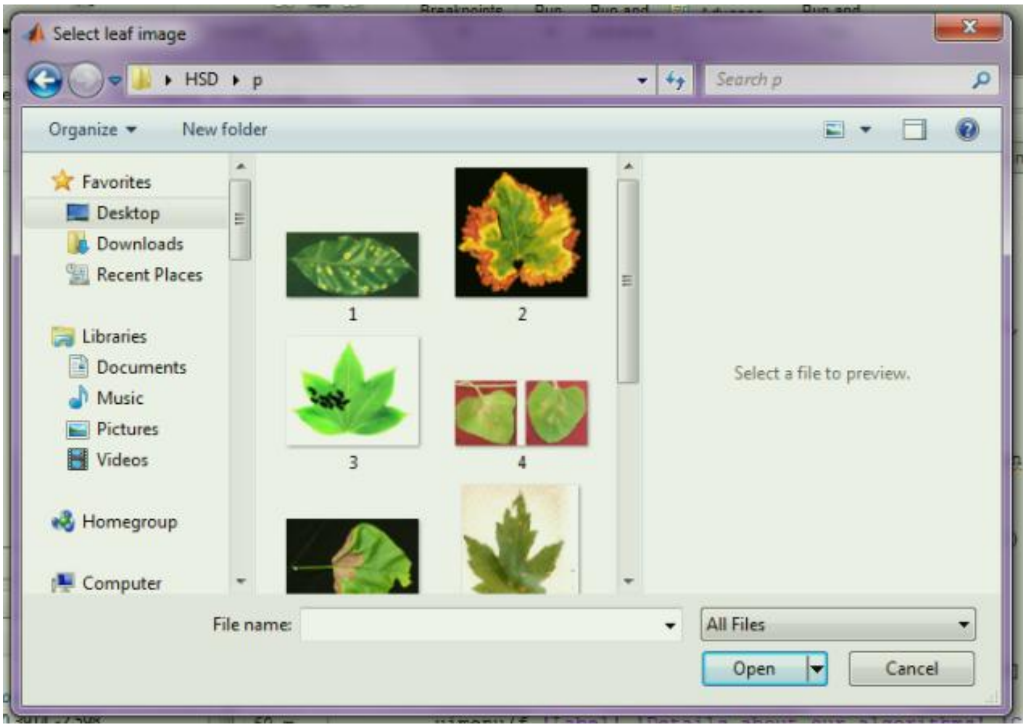


Fig-8

6. Select terminals of Defected leaf

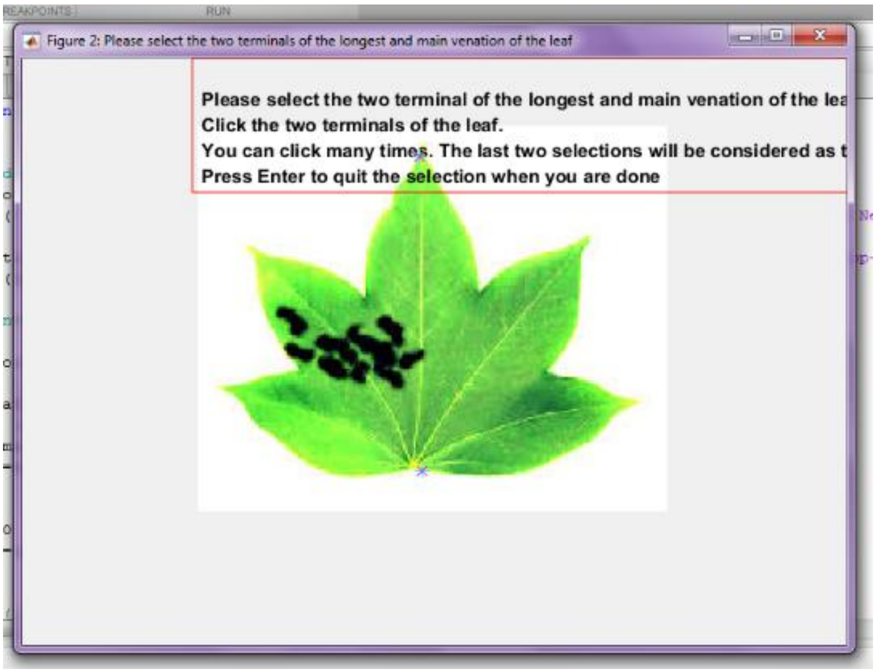


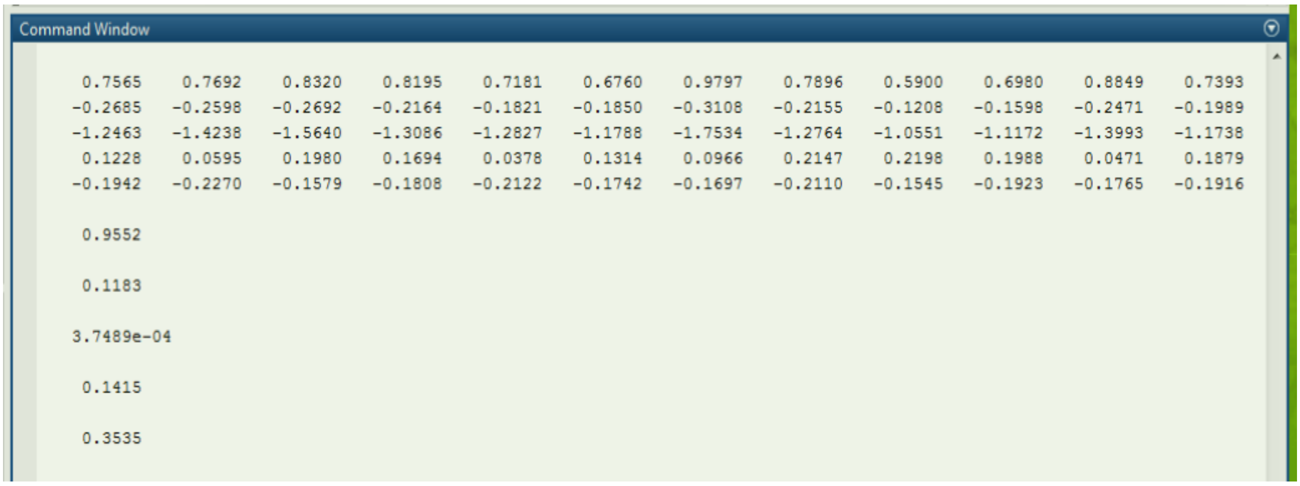
Fig-9

7. All the Training Value for Defected leaf

```
Command Window
  -0.1705   -0.1977   -0.2073   -0.1379   -0.1920   -0.1729   -0.1506   -0.1773   -0.1571   -0.1421   -0.1683   -0.2107

 Columns 1777 through 1788

   0.8371    0.7806    0.7115    0.9574    0.7377    0.8146    0.7576    0.6243    0.8533    0.7421    0.9035    0.7308
  -0.2513   -0.2152   -0.2267   -0.4687   -0.1778   -0.2267   -0.2034   -0.1092   -0.3023   -0.1836   -0.3256   -0.2063
  -1.4485   -1.2598   -1.2262   -1.9693   -1.1347   -1.3185   -1.2833   -0.9951   -1.5580   -1.3804   -1.6739   -1.4322
   0.0865    0.1385    0.0184   -0.2174   -0.0172    0.1137    0.1467    0.1702    0.2175    0.0671    0.1418    0.1627
  -0.2246   -0.1975   -0.2055   -0.2023   -0.2611   -0.1590   -0.1978   -0.2173   -0.1643   -0.2237   -0.1003   -0.1225

 Columns 1789 through 1800

   0.7565    0.7692    0.8320    0.8195    0.7181    0.6760    0.9797    0.7896    0.5900    0.6980    0.8849    0.7393
  -0.2685   -0.2598   -0.2692   -0.2164   -0.1821   -0.1850   -0.3108   -0.2155   -0.1208   -0.1598   -0.2471   -0.1989
  -1.2463   -1.4238   -1.5640   -1.3086   -1.2827   -1.1788   -1.7534   -1.2764   -1.0551   -1.1172   -1.3993   -1.1738
   0.1228    0.0595    0.1980    0.1694    0.0378    0.1314    0.0966    0.2147    0.2198    0.1988    0.0471    0.1879
  -0.1942   -0.2270   -0.1579   -0.1808   -0.2122   -0.1742   -0.1697   -0.2110   -0.1545   -0.1923   -0.1765   -0.1916

   0.9552

   0.1183
```

Fig-10

8. Output Result in terms of value. Here, negative values show the name of disease mentioned in documentation.



```
Command Window
   0.7565    0.7692    0.8320    0.8195    0.7181    0.6760    0.9797    0.7896    0.5900    0.6980    0.8849    0.7393
  -0.2685   -0.2598   -0.2692   -0.2164   -0.1821   -0.1850   -0.3108   -0.2155   -0.1208   -0.1598   -0.2471   -0.1989
  -1.2463   -1.4238   -1.5640   -1.3086   -1.2827   -1.1788   -1.7534   -1.2764   -1.0551   -1.1172   -1.3993   -1.1738
   0.1228    0.0595    0.1980    0.1694    0.0378    0.1314    0.0966    0.2147    0.2198    0.1988    0.0471    0.1879
  -0.1942   -0.2270   -0.1579   -0.1808   -0.2122   -0.1742   -0.1697   -0.2110   -0.1545   -0.1923   -0.1765   -0.1916

   0.9552

   0.1183

  3.7489e-04

   0.1415

   0.3535
```

Fig-11

INSTALLATION INSTRUCTIONS

From ISO

First, mount the .iso file of the image with right-click, Open with and select Disk image mounter (if required). Open a terminal and change to the mounted image file. For example:

cd /media/username/MATHWORKS_R2014A

Install

Run the installer as a super user
sudo ./install

Follow the instructions of the installer and install in /usr/local/MATLAB/R2014a To add an entry to the launcher and some other features, install the matlab-support package:

sudo apt-get install matlab-support

During the installation you have to confirm the installation path of matlab and could restrict the installation only to certain users (which I did not require). In addition you could rename the GCC libraries, but I had no problems so far so I didn't deactivate those. After completion, you have to change the ownership of the .matlab folder in the home folder to your own user account if you do not want to run Matlab as a super user.

cd ~ sudo chown username -R ~/.matlab

Now Matlab should also be launchable through unity and should work as intended.

Chapter 6

CONCLUSION AND FUTURE SCOPE

This report introduces a neural network approach for plant leaf recognition. The computer can automatically classify some kinds of plants via the leaf images loaded from digital cameras or scanners. PNN adopted has fast speed on training and simple structure. Compared with other methods, this algorithm is fast in execution, efficient in recognition and easy in implementation. The experimental results indicate that the proposed approach is a valuable approach, which can significantly support an accurate detection of leaves in little computational effort.

In the present scenario it is very important to have an established approach for grading leaves automatically. For this, a system based on Machine Vision Technology and Artificial Neural Network (ANN) is of great use for automatically detecting the leaf. These systems are going to be very helpful for agriculturist since it is efficient than the manual method. These systems can be used to replace the manual leaf recognition technique and can be used by agricultural experts in identifying correct pesticide and its quantity to overcome the problem in an efficient and effective manner.

REFERENCES

[1] Wu, Stephen Gang, et al. "A leaf recognition algorithm for plant classification using probabilistic neural network." Signal Processing and Information Technology, 2007 IEEE International Symposium on. IEEE, 2007.

[2] Al-Hiary H., S. Bani-Ahmad, M. Reyalat, M., Braik and Z. Al Rahamneh, “Fast and accurate detection and classification of plant disease”, International Journal of computer Application, ISSN 0975-8887, vol.17, No1, pp. 31-38, 2011

[3] Huang, Guang-Bin, et al. "Extreme learning machine for regression and multiclass classification." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42.2 (2012): 513-529.

[4] Mao, Ke Zhi, K-C. Tan, and Wee Ser. "Probabilistic neural-network structure determination for pattern classification." Neural Networks, IEEE Transactions on 11.4 (2000): 1009-1016.

[5] Wang, Haiguang, et al. "Image recognition of plant diseases based on backpropagation networks." Image and Signal Processing (CISP), 2012 5th International Congress on. IEEE, 2012.

[6] Rastogi, Aakanksha. "Leaf Disease Detection and Grading using Computer Vision Technology & Fuzzy Logic.” Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on Year: 2015.

[7] Li, YunFeng, et al. "A leaf vein extraction method based on snakes technique."Neural Networks and Brain, 2005. ICNN&B'05. International Conference on. Vol. 2. IEEE, 2005.

[8] Al Bashish, Dheeb, Malik Braik, and Sulieman Bani-Ahmad. "Detection and classification of leaf diseases using K-means-based segmentation and neural-networks-based classification." Information technology journal 10.2 (2011): 267-275.

[9] Gouveia, F., et al. "Biometry: the characterization of chestnut-tree leaves using computer vision." Industrial Electronics, 1997. ISIE’97. Proceedings of the IEEE International Symposium on. IEEE, 1997.

[10] Hong, Sung Min, Bruce Simpson, and Gladimir VG Baranoski. "Interactive venation-based leaf shape modeling." Journal of Visualization and Computer Animation 16.3-4 (2005): 415-427.

[11] Rashad, M. Z., B. S. El-Desouky, and Manal S. Khawasik. "Plants images classification based on textural features using combined classifier."International Journal of Computer Science and Information Technology 3.4 (2011): 93-100.

[12] H. G. Wang, Z. H. Ma, M. R. Zhang, and S. D. Shi, “Application of computer technology in plant pathology”, Agriculture Network Information, vol. 19, pp. 31–34, October 2004.

[13] Y. Li, Q. Zhu, Y. Cao, and C. Wang, “A leaf vein extraction method based on snakes technique,” in Proceedings of IEEE International Conference on Neural Networks and Brain, 2005.

[14] H. Fu and Z. Chi, “Combined thresholding and neural network approach for vein pattern extraction from leaf images,” IEE Proceedings-Vision, Image and Signal Processing, vol. 153, no. 6, December 2006.

[15] S. M. Hong, B. Simpson, and G. V. G. Baranoski, “Interactive venationbased leaf shape modeling,” Computer Animation and Virtual Worlds, vol. 16, 2005.