ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Comparison of Leaf Recognition using Multi-layer Perceptron and Support Vector Machine

[1] Juby George, [2]Gladston Raj S
[1] Research Scholar School of Computer Science, Mahatma Gandhi University ,Kottayam
[2] Head, Department of Computer Science, Govt. College Nedumangad, Thiruvananthapuram

Abstract: Identification of leaves from digital images using various automatic pattern recognition algorithms results in performance degradations. Here, various leaf features are been extracted and exposed to Multi-layer Perceptron and Support Vector Machine. The leaf images are taken from the Columbia Dataset and that are preprocessed to get the region of interest (ROI). The shape, color and vein features are then extracted from the selected ROI. The prominent features are then found out by using Principal Component Analysis (PCA). The reduced feature sets are supplied to these algorithms for identification. Total 150 samples were taken from this dataset spreads over 10 different leaf species. Multilayer Perceptron and Support vector machine are trained with 104 leaf images and are been tested and validated using 23 leaf images each. A comparison is made between the performances of these two learning methods and found that the leaf recognition accuracy of the support vector machine is better than that of Multilayer Perceptron algorithm.

Index Terms — Feature extraction, Shape features, Vein features, Wiener filter, Color features, Principal Component Analysis, Multi-layer feed forward network, Support Vector Machine.

INTRODUCTION

There have been many recent studies on plant identification to recognize plant species. One way to recognize the plant is from their leaves because every plant has a unique leaf [1]. The feature extraction step can easily transfer the leaf image to a computer and then the computer can extract necessary features automatically using image processing techniques and recognize the plant leaf using machine learning.

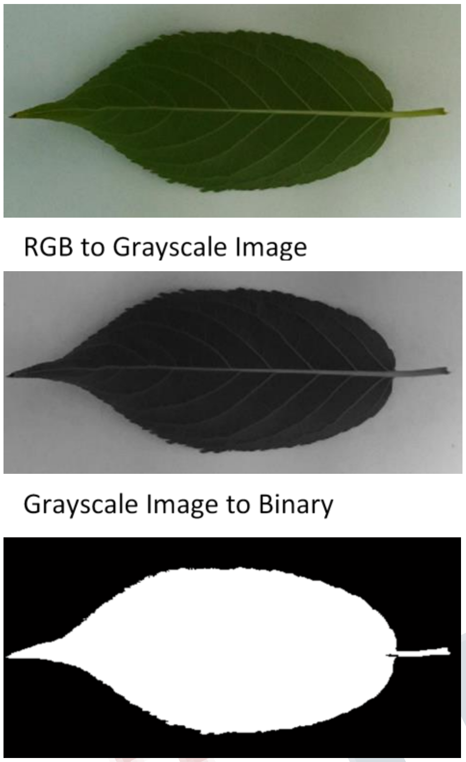
The Support Vector Machine (SVM) was first proposed by Vapnik and has since attracted a high degree of interest in the machine learning research community. Several recent studies have reported that the SVM (support vector machines) generally are capable of delivering higher performance in terms of classification accuracy than the other data classification algorithms[4][5]. SVM has been employed in a wide range of real world problems such as text categorization, hand-written digit recognition, tone recognition, image classification and object detection, micro-array gene expression data analysis, data classification [2]. For any datasets, the performance of SVM is very sensitive to how the cost parameter and kernel parameters are set [3][4][5].

It is well known that the correct way to extract plant features involves plant recognition based on leaf images[7]. The features which are widely used for plant recognition based on leaf images are color and shape[7]-[13]. In the color-based conventional study, a simple color similarity between two images can be measured by comparing their color histogram [7]. Some systems employ descriptions used by botanists[13]. This work automatically extracts colour, shape and vein features after preprocessing of the leaf images. PCA is used for identifying the prominent features from shape, color and vein features. Multi-layer perceptron and Support Vector Machine are used for recognizing the leaf, based on the selected leaf features. The algorithm for leaf recognition system is as follows

Step 1: Select leaf images from database
Step 2: Preprocess and extract ROI.
Step 3: Feature extraction: shape, color and vein.
Step 4: Perform PCA for selecting principal features
Step 5: Apply Multilayer Perceptron on selected feature set.
Step 6: Apply SVM for Leaf recognition
Step 7: Comparing the Performance of both

II. MAGE PREPROCESSING

Leaf Images are taken from Columbia Leaf Image Database. An RGB image is first converted to gray scale image. Boundary enhancement of leaf images are performed using morphological operations and sobel edge filtering thus extracts the region of interest from background. An example of image pre-processing is illustrated in Fig.2.



RGB to Grayscale Image

Grayscale Image to Binary

Fig 2 : A preprocessing example.

III. LEAF FEATURE EXTRACTION

A. SHAPE FEATURE

1) Diameter: The diameter is defined as the longest distance between any two points on the boundary of the leaf. It is denoted as D[10][13].
2) Physiological width: Drawing a line passing through the two terminals of the main vein can plot infinite line orthogonal to that line. The longest distance between points of those intersection pairs is defined at the physiological width. It is denoted as Wp [13].
3) Physiological length: The distance between two terminals of the main vein of the leaf is defined as the physiological length. It is denoted as Lp [13].
4) Leaf area: The value of leaf area is easy to evaluate, just counting the number of pixels of binary value 1 inside the boundary of leaf on preprocessed leaf image. It is denoted as A[13].
5) Leaf perimeter: Denoted as P, calculated by counting the number of pixels on the boundary consisting leaf margin[13].
6) Convex Area: The number of pixels in convex image[7].
7) Eccentricity: The ratio of the distance between the foci of the ellipse and its major axis length[7].
8) Solidity: The proportion of the pixels in the convex hull that are also in the region. Computed as area/convex area[7].
9) Major axis length: The length of the major axis of the ellipse that has the same second-moments as the region[7].

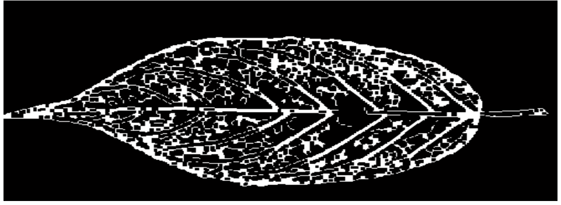
10) Minor axis length: The length of the minor axis of the ellipse[2].

B. LEAF COLOR FEATURES

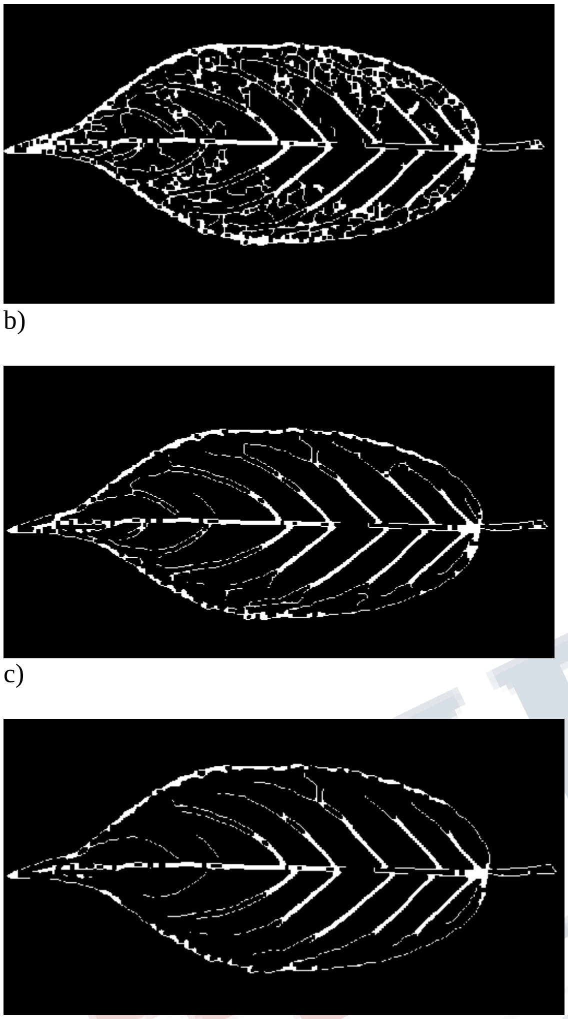
Color moments are measures that can effectively use to discriminate images based on their features. Color moments are very important and helpful to distinguish color based features. The information can be extracted by using low order moments. The four color features extracted are for Red ,Green and Blue colors:
1) Mean[1]
2) Standard Deviation[1]
3) Skewness[1]
4) Kurtosis[1]

C. VEIN FEATURE EXTRACTION USING WIENER FILTER

Following steps are used for extracting the leaf vein feature:
1) Perform image pre-processing[6] and extracting the region of interest[6].
2) Convert the image to grayscale and extract gray level co-occurrence matrix[6].
3) Perform wiener filtering by assigning the threshold[8] value by computing average gray level value from the gray level histogram[8].
4) Perform Disk structuring element using canny edge function[12].
5) Compute the number of white pixels and taking the ratio vein details at different levels are measured v1,v2,v3,v4,v41
The different vein structure at each type is illustrated in Fig.3.



a)



b)

c)

d)

Fig 3: a) Vein 1(V1) b) Vein2(V2) c) Vein3(V3) d) Vein4(V4)

IV. PROPOSED SCHEME

A. Principal Component Analysis (PCA)

PCA is used to reduce the number of features and the dimension of input vector of neural network[10]. The ranking of features are performed using Rankers Algorithm.

In this work, the contribution of five principal components are selected from ten shape features, six principal features are identified out of twelve color features and three principal vein features are selected from five vein features. These 14 features are used as the input vectors of neural network and SVM.

B. Introduction to multi-layer feed forward network

An artificial neural network is an interconnected group of artificial neurons simulating the thinking process of human brain[8]. The multi-layer feed forward neural network[13] is the workhorse of the neural network. It can be used for both function fitting and pattern recognition problems. Pattern recognition is the process of training a neural network to assign the correct target classes to a set of input patterns[11]. Once trained, the network can be used to classify patterns it has not seen before. A multi-layer feed forward neural network consists of input nodes, one or more number of hidden nodes, and one output layer[13].

C. Training the Network

Total 150 leaves from Columbia leaf set database are taken as samples. 150 leaves represent 10 different leaf species. 14 different principal variables are identified using PCA and is used for training the network. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation[8] set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error[8]. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. The network weights and biases are saved at the minimum of the validation set error. Out of 150 samples 104 samples are used as training data and 23 samples are used as validation and test data.

Once the network weights and biases are initialized, the network was ready for training. The multi-layer feed forward network is trained for pattern recognition[13]. The training process requires a set of examples of proper network behavior network inputs p and target outputs t.

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function[15]. The default performance function for feed forward networks is mean square error Equ.8, the average squared error between the networks outputs a and the target outputs t. It is defined as follows[14]:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad [14]$$

(1)

Training function used for pattern recognition networks are trainscg[14]: Scaled Conjugate Gradient. Scaled

Conjugate Gradient (trainscg) does not require line search at each iteration step like other conjugate training functions. Step size scaling mechanism is used which avoids a time consuming line search per learning iteration. This mechanism makes the algorithm faster than any other second order algorithms[15].

The memory requirements are relatively small, and much faster than standard gradient descent algorithms. The magnitude of the gradient and the number of validation checks are used to terminate the training[14]. The gradient will become very small as the training reaches a minimum of the performance. If the magnitude of the gradient is less than 1e-5, the training will stop.

D. Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes two different classes.

In Figure there are many hyper planes which can be fit in to classify the data but which one is the best is the right or correct solution. The need for SVM arises. (Taken Andrew W. Moore 2003) [1]. Note the legend is not described as they are sample plotting to make understand the concepts involved.

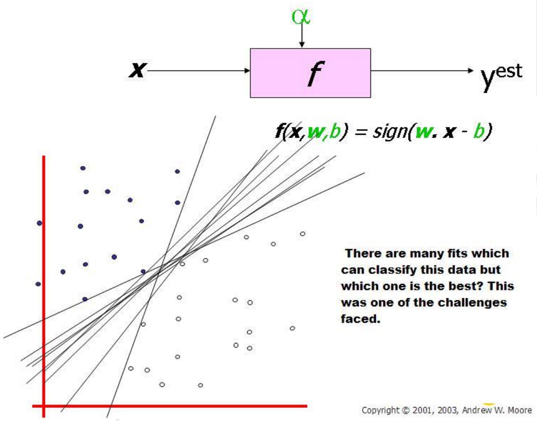

α
x
f
y^est
f(x,w,b) = sign(w. x - b)
There are many fits which can classify this data but which one is the best? This was one of the challenges faced.
Copyright © 2001, 2003, Andrew W. Moore

Fig 4: Many hyper planes to classify the data

There are many linear classifiers (hyper planes) that separate the data. However only one of these achieves maximum separation. The reason for that is by considering a hyper plane to classify; the classification might end up closer to one set of datasets compared to others. This can be avoid by considering the concept of maximum margin classifier or hyper plane as an apparent solution. The next illustration gives the maximum margin classifier example which provides a solution to the mentioned problem [5].

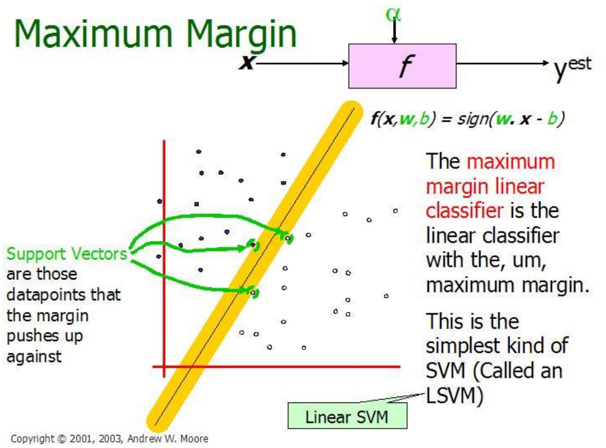

Maximum Margin
α
x
f
y^est
f(x,w,b) = sign(w. x - b)
The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)
Support Vectors are those datapoints that the margin pushes up against
Linear SVM
Copyright © 2001, 2003, Andrew W. Moore

Fig 5: Illustration of Linear SVM

Expression for Maximum margin is given as [4][5]

$$\text{margin} \equiv \underset{\mathbf{x} \in D}{\arg\min}\, d(\mathbf{x}) = \underset{\mathbf{x} \in D}{\arg\min} \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

The goal of SVM is separating the data with hyper plane and extends this to non-linear boundaries using kernel trick [4] [5]. Considering maximum margin helps to avoid local minima and better classification. For mathematically,

$\mathrm{wx}_i + \mathrm{b} \geq 1$ [i] If $\mathrm{Y_i} = +1$; (2)
[ii] If $\mathrm{Y_i} = -1$; $\mathrm{wx_i} + \mathrm{b} \leq 1$
[iii] For all i; $\mathrm{y_i}\,(\mathrm{w_i} + \mathrm{b}) \geq 1$

In this equation x is a vector point and w is weight. So to separate the data [a] should always be greater than zero. Among all possible hyper planes, SVM selects the one where the distance of hyper plane is as large as possible. If the training data is good and every test vector is located in radius r from training vector. Now if the chosen hyper plane is located at the farthest possible from the data [12]. This desired hyper plane which maximizes the margin also bisects the lines between closest points on convex hull of the two datasets and have options[i], [ii] & [iii].

Distance of closest point on hyper plane to origin can be found by maximizing the x, as x is on the hyper plane. Similarly for the other side points also have a similar scenario. Thus solving and subtracting the two distances results the summed distance from the separating

hyperplane to nearest points[2]. Maximum Margin = M = 2 / ||w||
The quadratic optimization problem then will be used to solve w and b. The problem is solved by optimizing the quadratic function with linear constraints. The solution involves constructing a dual problem and where a Langlier’s multiplier αi is associated. To find w and b such that Φ (w) =½ |w’||w| is minimized[2];
And for all {(xi, yi)}: yi (w * xi + b) ≥ 1.
Now solving: w =Σαi * xi; b= yk- w *xk for any xk such that αk□ 0
Now the classifying function will have the following form: f(x) = Σαi yi xi * x + b
(3)

Figure 6: Representation of Support Vectors (Copyright © 2003, Andrew W. Moore)[1]

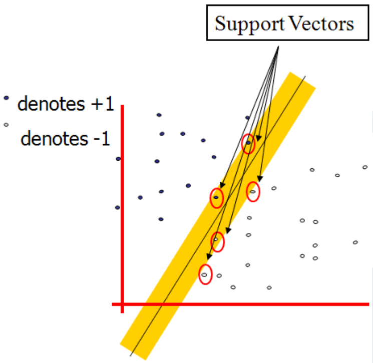


Fig 6 : Representation of Support Vectors

Kernel: If data is linear, a separating hyper plane may be used to divide the data. However it is often the case that the data is far from linear and the datasets are inseparable. To allow this kernels are used to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable [1]. A very simple illustration of this is shown below in figure 7 [3] [4] [5].

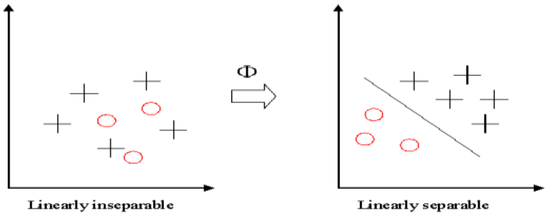


Fig 7: Why Kernels?

This mapping is defined by the Kernel:

$$K(x,y) = \Phi(x) \cdot \Phi(y) \quad (4)$$

Feature Space: Transforming the data into feature space makes it possible to define a similarity measure on the basis of the dot product. If the feature space is chosen suitably, pattern recognition can be easy [1].

$$\langle x_1 \cdot x_2 \rangle \leftarrow K(x_1,x_2) = \langle \Phi(x_1) \cdot \Phi(x_2) \rangle \quad (5)$$

In this work, the kernel used is polynomial kernels because the classification of leaf feature dataset performed well by using Polynomial kernel.
Polynomial: A polynomial mapping is a popular method for non-linear modeling[2].

$$K(x,x') = \langle x,x' \rangle^d.$$

$$K(x,x') = (\langle x,x' \rangle + 1)^d. \quad (6)$$

V. EXPERIMENTAL RESULT

By comparing the Multilayer Perceptron and Support vector machine, the performance of Support vector Machine was good by using Polynomial Kernel. Each color, shape and vein features selected after the principal component analysis is tested and the features are tested together. Ten species of Plant leaves with reduced feature set was taken for recognition by MLP and SVM and 70% leaf features are used for training and remaining 15% data used for validation and 15% used for testing.

Percentage of leaf recognition success rate is listed in the Table 1. The success rate of the leaf recognition was compared by first classified the leaves by color features and then by shape features and vein features separately. The result of Support vector machine approach produced 71.15% success rate for color features, 73.07% success rate for shape features and 66.34% success rate for vein features. By combining color features and shape features the success rate is 91.34% and the success rate of reduced features by combining three categories, shape, color and vein is 96.13%.

TABLE I. RESULT

Sl. NO :	Success Rate Comparison		
	Features	***% of Recognition (MLP)***	***% of Recognitio n (SVM)***
1	Color Features	63.3%	71.15%
2	Shape Features	69.3%	73.07%

Sl. NO :	Success Rate Comparison		
	Features	***% of Recognition (MLP)***	***% of Recognitio n (SVM)***
3	Vein Features	57.3%	66.34%
4	Color Features+ Shape Features	84.7%	91.34%
5	Color Features+ Shape Features+ Vein Features	95.3%	96.13%

VI. CONCLUSION

This paper describes a comparative study of MLP and SVM approach for plant leaf recognition. This system automatically recognizes ten types of plant leaves by extracting the features and selecting the principle features by PCA analysis. Multilayer perceptron (MLP) properties include universal approximation of continuous nonlinear functions. In Multilayer Perceptron the network architectures had multiple inputs and outputs and the numbers of hidden layers were determined by the performance through trial and by avoiding the chance to overfit the data. Some of them having many local minima and also finding how many neurons might be a task. Another issue with MLP is to determine whether optimality of that NN is reached. Even if the neural network solutions used are tends to converge, this may not result in a unique solution. ANNs often overfit if training goes on too long, that means for any given pattern, an ANN might start to consider the noise as part of the pattern.

SVM performed well compared to Multilayer Perceptron. In leaf recognition polynomial kernel recognize the leaf classes accurately by having a high performance comparing to MLP. The major strengths of SVM are the training is relatively easy because of no local optimal solutions. It scales relatively well to high dimensional data. The trade-off between classifier complexity and error can be controlled explicitly.

REFERENCES

[1] V. Patil and R. R. Manza, “A method of feature extraction from leaf architecture,” IJARCSSE, vol. 5, no. 7, Jul. 2015.

[2] Tutorial Slides by Andrew Moore. http://www.cs.cmu.edu/~awm.

[3] Vikramaditya Jakkula, Tutorial on Support Vector Machine (SVM) School of EECS, Washington State University, Pullman 99164

[4] Burges C., “A tutorial on support vector machines for pattern recognition”, In “Data Mining and Knowledge Discovery”. Kluwer Academic Publishers, Boston, 1998, (Volume 2).

[5] Nello Cristianini and John Shawe-Taylor, “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods”, Cambridge University Press, 2000.

[6] R. . Gonzalez, R. . Woods, and S. . Eddins, Digital image processing using MATLAB. Prentice Hall.

[7] K.-B. Lee and K.-S. Hong, “An implementation of leaf recognition system using leaf vein and shape,” International Journal of Bio-Science and Bio-Technology, vol. 5, no. 2, pp. 57–66.

[8] J. Mounsef and L. Karam, “Fully automated quantification of leaf venation structure,” presented at the 2012 International Conference on Artificial Intelligence, ICAI 2012, 2012.

[9] G. Panchal, A. Ganatra, and D. Panchal, “Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers,” International Journal of Computer Theory and Engineering, vol. 3, no. 2, Apr. 2011.

[10] K. Singh, I. Gupta, and S. Gupta, “SVM-BDT PNN Fourier Moment Technique for Classification of Leaf Shape,” International Journal of Signal Processing,Image Processing and Pattern Recognition, vol. 3, no. 4, Dec. 2010.

[11] D. F. Specht, “Probabilistic neural networks,” vol. 3, no. 1, pp. 109–118, 1990.

[12] V. Tajane and N. Janwe, “Medicinal plants disease identification using canny edge detection algorithm, histogram analysis and CBIR - Semantic Scholar,” IJARCSSE, vol. 4, no. 6, Jun. 2014.

[13] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A Leaf recognition algorithm for plant classification using probabilistic neural network," arXiv:0707.4289 [cs], Jul. 2007.

[14] "Multilayer neural network architecture - MATLAB & Simulink - MathWorks India," in.mathworks.com, 2017. [Online]. Available: http://in.mathworks.com/help/nnet/ug/multilayer-neural-network-architecture.html. [Accessed: 04-Jun-2017].

[15] "Classify patterns with a neural network - MATLAB & Simulink - MathWorks India." [Online]. Available:https://in.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html. [Accessed: 04-Jun-2017].