Multimedia Tools and Applications (2019) 78:27785–27808
https://doi.org/10.1007/s11042-019-7588-2

Multiclass Twin Support Vector Machine for plant species identification

Neha Goyal[1] · Kapil Gupta[1] · Nitin Kumar[2]

Received: 7 June 2018 / Revised: 5 February 2019 / Accepted: 3 April 2019 /
Published online: 26 June 2019

Abstract

Automatic plant species identification is one of the recent and fascinating research area as plants are crucial element of ecosystem. Several plant species exist with significant importance but most of us are unaware of the diversity of plant species available on earth. Their utility to humans starts as oxygen provider, food source, and medicinal compounds essential for medicines that are difficult to develop in right proportions. Being the first living habitants of earth, they have roots far deeper in the ecosystem than any living being. Hence, it is utmost important to develop automatic plant species identification system in which the digital image of the plant is given as input and the label of the plant is determined by the system. In this paper, we have focused on three different aspects (i) Significance of threshold (ii) Feature descriptor that can best describe the leaf images and (iii) Proposed a novel classification method called Multi class Twin Support Vector Machine which in an extension of widely used Twin Support Vector Machine classifier. The performance of the proposed method is compared with SVM, Multi Birth SVM and Probabilistic Neural Network. It is observed that the proposed classifier outperforms all the aforementioned classifiers on publicly available datasets.

Keywords Plant recognition · Image segmentation · Feature extraction · Multiclass classification · TWSVM

✉ Neha Goyal
neha.goyal2309@gmail.com

Kapil Gupta
navkapil@gmail.com

Nitin Kumar
nitin@nituk.ac.in

1 NIT, Kurukshetra, India

2 NIT, Uttarakhand, India

1 Introduction

Plant stratification is a scientific and systematic approach for plant classification. It is laborious and time-consuming task as well as a challenging problem due to taxonomic properties of the plants [10, 20, 33]. A semi-automatic plant stratification system needs cataloguing of plant species. The system can help aware professionals, enthusiasts, researchers and other interested people about the diversity of the plants, their taxonomic properties. Further, it promotes people to become citizen scientist [10, 17, 28]. The most natural user interaction method for an unaware plant enthusiast is possible through plant images. The information can be rapidly disseminated by automated plant species identification. We need an efficient system that takes visual data of a plant leaf, extract various features and label it to corresponding plant species. Image based plant recognition system is a promising step. In manual identification, botanists use all available properties of the plant like leaf, flower, fruit, stem and seeds while most of research work for identification using digital images is based on leaf as they are numerous and remain throughout the year [19, 33]. Earlier, algorithms were developed that take taxonomic properties collected by botanists manually to classify plants using computational analysis but it is found to be difficult to transfer data to computer system automatically. In last few decades, researchers have put efforts to automate the system by extracting numerous features using leaf images and classification methods [6, 7, 22, 29, 30]. Wu et al. [31] proposed an algorithm for plant identification system. They used 12 morphological features based on 5 basic geometrical features. These 12 features are reduced to five features using Principal Component Analysis (PCA) and Probabilistic Neural Network (PNN) is used to classify the leaves. Aakif and Khan [1] used morphological features, Fourier descriptor and proposed shape defining feature. Extracted features are fed to artificial neural network(ANN) for classification. Kalyoncu and Toygar [13] used two databases flavia [25] and leafsnap [15] and used adaptive threshold segmentation for flavia dataset over blue channel. They used moment invariants, perimeter ratio, convexity and multi-scale distance matrix as feature vector. Linear discriminant classifier was used for classification purpose. Du et al. [11] used their own database collected in lab by scanning leaves and extracted morphological features and 7 hu invariant moments. They proposed Move Median Centre (MMC) hyper-sphere classifier for leaf recognition and compared the results with K-Nearest Neighbour (NN) classification algorithm. In recent years, plant recognition system has attracted a lot of researchers who have proposed several feature extraction methods.

There are number of classification algorithms like ANN, PNN, SVM that are used for plant species identification. Cortes and Vapnik [8] proposed SVM, a binary classifier based on maximum margin classification function to adjust the capacity of the model to the size and complexity of the training data. Twin SVM (TWSVM) [14], a binary classifier that determines two non-parallel planes by solving two related problems of type SVM. SVM and TWSVM are both extended from binary classification approach to multi class classification approach using one v/s one or one v/s rest. Least Square Twin Support Vector Machine [27], a binary classifier is also an extended version of binary TWSVM using formulation of one v/s one, one v/s all, all v/s one and directed acyclic graph. Yang et al. [32] proposed Multi Birth SVM (MBSVM) with lower computational complexity by solving k Quadratic Programming Problems (QPP's) for k-class classification problem. In this article, we focused on three objectives- (i) significance of threshold to convert gray scale image to binary images to extract shape features (ii) comparative analysis of different feature descriptor to find feature set that can best describe the leaf images and (iii) multi class classification algorithm based on TWSVM has proposed to identify the plant species from leaf image. The proposed classifier is the extension of TWSVM that tends to find k- hyperplane for each class by solving k smaller size QPP's. Each hyperplane is as closer as possible to one of the k classes and at least a unit distance far from other classes. The effectiveness of the proposed classifier is confirmed by comparing the results with Wu et. al. for thresholding, HOG, LBP for feature sets and methods available in literature such as PNN, SVM and MBSVM.

The rest of paper is organized as follows: Section 2 provides the background information for the proposed method. In Section 3, the proposed method is described in detail. Experimental setup and results are given in Section 4. Conclusion and future work is included in Section 5 at the end.

2 Background

Plant species identification is a systematic and sequential procedure. It consists of three steps after image acquisition. In the first step, leaf segmentation is carried out. It deals with pre-processing of image to prepare the image for better morphological analysis. Pre-processing is essential step in identification process to reduce noise, background subtraction and content enhancement. In the next step, features are extracted based on shape, margin, vein and color etc. In leaf identification, Feature extraction aims to extract various features based on color, texture and shape. After feature extraction, best combinations of features are selected as a means to reduce the conditionality of the data to make classification better and to improve performance of recognition system. In classification process, extracted features are used to train the model and test the classification accuracy of the model. Figure 1 represents the framework of the plant identification system.

2.1 Image pre-processing

In the image preprocessing stage, the raw image I is RGB image that is firstly converted to gray scale image g using threshold segmentation technique as in (1) where R, G and B indicate the channels of a color pixel. Using (2), g is converted to binary image b.

$$g = R * 0.2989 + G * 0.5870 + B * 0.1140 \tag{1}$$

$$b(x, y) = \begin{cases} 1, & \text{if } g(x, y) \geq \text{ threshold.} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where x, y is the coordinate of pixel, $b(x, y)$ is the color intensity of (x, y) pixel in binary image either 0 or 1, $g(x, y)$ is the color intensity of (x, y) pixel in Gray scale image range

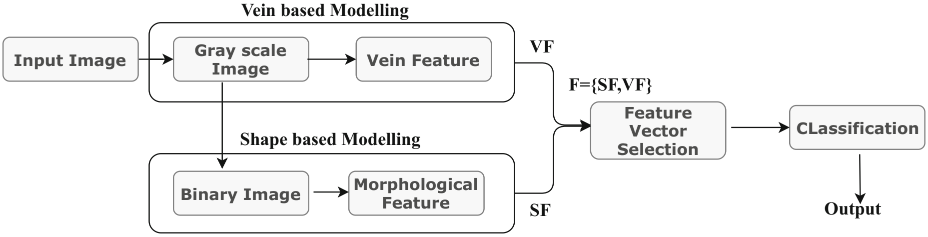

Fig. 1 Frame work of plant identification system

from 0-255 and $threshold$ is selected by average histogram [21, 24] method. Next, boundary extraction from the binary image is carried out. Firstly, an averaging filter of size 3 × 3 is used for noise reduction. Filtering is linear combination of neighbourhood pixel. Using averaging filter, each element in the image is replaced by mean of neighbourhood pixel.

For edge detection, convolution operation is performed using Laplacian filter with alpha value 0.2. A 3 × 3 laplacian filter is created using (3).

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3}$$

$$\nabla^2 = \frac{4}{\alpha + 1} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix} \tag{4}$$

Figure 2 represent the results of various operation performed to find leaf boundary from RGB color image. For black boundary of image with white background pixel value is converted from 1 to 0 and vice versa.

2.2 Feature extraction

After image segmentation, features are extracted to train the model. As there are plenty of information to represent the object but to extract relevant information that best describes the object, is a challenging task in any recognition system [5]. Segmented leaf is used to extract feature. In this paper, 7 morphological features and 5 vein features based on 5 basic geometrical features whereas 8 Hog features and 59 LBP features as texture feature descriptor [4] are extracted to define the leaf images.

Basic Geometrical Features

1. *Diameter(D):* Diameter is the longest distance of any two pixels on leaf boundary and it is denoted by D.
2. *Physiological Length (L_p):* It specifies the length of the major axis of the ellipse that has same normalized second moments.

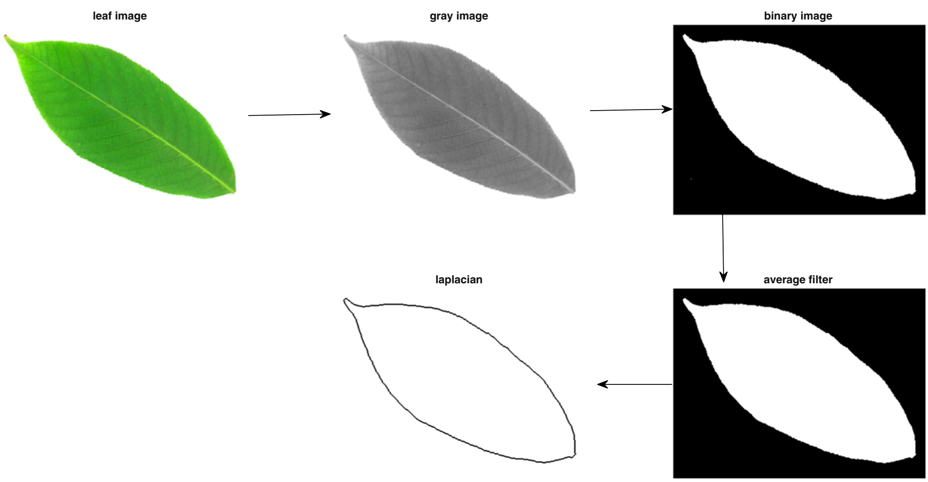

Fig. 2 Image pre-processing

3. *Physiological Width (W_p):* It specifies the length of the minor axis of the ellipse.
4. *Leaf Area:* Leaf area is total number of pixels equal to 1 in binary image after filtering and it is denoted as A.
5. *Leaf Perimeter:* Leaf perimeter, is denoted as P, is total number of pixels on the boundary of leaf image.

Shape based Morphological feature Morphology is the study of structural features and forms of some object based on shape. Morphological features of a leaf define internal structure of leaf. There are 7 morphological features discussed below which are calculated using basic geometrical features. Each leaf is represented as feature vector containing 7 elements of shape-based features

$$SF = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \end{bmatrix} \tag{5}$$

1. *Smooth Factor (s_1):* Effect of noise is used to represent smoothness of leaf when more than one filter of different size is applied. It is the ratio of leaf area smoothed using average filters of size 5×5 to the leaf area calculated after smoothing with average filter of size 2x2
2. *Aspect Ratio (s_2):* It is ratio of physiological length to physiological width.

$$AR = \frac{L_p}{W_p} \tag{6}$$

3. *Form Factor (s_3):* Difference between the leaf and circle is used as form factor

$$Form\_F = \frac{4 \times \pi \times A}{P^2} \tag{7}$$

4. *Rectangularity (s_4):* It is similarity between the leaf area and enclosing rectangle

$$Rectangularity = \frac{L_p \times W_p}{A} \tag{8}$$

5. *Narrow Factor (s_5):* It is defined as ratio of Diameter(D) to Physiological length (L_p)

$$Narrow\_F = \frac{D}{L_p} \tag{9}$$

6. *Ratio of Perimeter to Diameter (s_6):*

$$P\_D = \frac{P}{D} \tag{10}$$

7. *Perimeter ratio of physiological length and physiological width (s_7):* It is the ratio of perimeter to sum of physiological length and physiological width.

$$P\_LW = \frac{P}{L_p + W_p} \tag{11}$$

Vein Features Morphological opening on gray scale image is performed using structural element of different shape and subtracted from leaf margin. Resultant image will be veins of leaf.

$$v_1 = \frac{av_1}{A}, v_2 = \frac{av_2}{A}, v_3 = \frac{av_3}{A}, v_4 = \frac{av_4}{A}, v_5 = \frac{av_4}{av_1}$$

where av_r stands for area of vein and r indicates the radius of structural element of disk type and $r = 1, 2, 3, 4$. Finally, 5 vein features are the ratio of vein's area corresponding to different radius and leaf area.

$$VF = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix} \tag{12}$$

Histogram of oriented gradient Histogram of Oriented Gradient(HOG)is based on concept of local shape and appearance of an object with in an image. Dalal and triggs [9] represents human detection algorithm with better performances that utilized the counts of occurrences of gradient orientation in localized portion of an image.

Local binary pattern Local Binary Pattern (LBP) [4] is powerful texture analysis technique based on local spatial content of an image. It is widely used concept for labelling the pixel by thresholding using center pixel and taking the difference of neighbour pixels. Finally, summing the threshold values as binary numbers.

2.2.1 Combined features

A leaf is characterized and recognized by its shape and vein features. Shape feature may not helpful in recognizing plant species having similar shape or same geometrical features. To make combined feature set, 3 different type of feature sets are generated. Shape-based feature vector is concatenated with vein-based feature vector, HOG and LBP as given below:

$$F_1 = \left[\, SF \;\; VF \,\right] \qquad F_2 = \left[\, SF \;\; HOG \,\right] \qquad F_3 = \left[\, SF \;\; LBP \,\right] \tag{13}$$

2.3 Principal Component Analysis (PCA)

PCA [2] is machine learning algorithm for dimensionality reduction, noise reduction, or to compress data size. PCA analyse the structural information to extract important information from it. It approached to find principal component based on highest eigen value to map d-dimensional data into $n(< d)$ -dimensional dataset that is linear combination of original dataset. Eigen vector corresponding to highest eigen value also known as principal components. In this paper, PCA is used to reduce the dimensionality of three feature sets (F_1, F_2 and F_3) to 5,5 and 8 number of features respectievly.

2.4 Popular classification methods

In this section, popular classification algorithms are included that have shown promising results with plant species identification such as PNN, SVM, and some variants of SVM. TWSVM [14] is binary class classification approach that seeks to find two non-parallel hyperplanes such that each hyperplane is closer to one of the two classes and at least unit distance far away from other class. Our proposed classifier is extension of TWSVM that tends to find k hyperplanes, one for each class by solving k smaller size QPP's.

2.4.1 Preliminaries

Here, some notations are provided for better understanding and smoothness of the article. The training dataset has m data points.

$$T = \{(x_1, y_1), (x_2, y_2) ... (x_m, y_m)\} \tag{14}$$

where $x_l \in R^n, \forall l = 1, 2, ..., m$ are m-dimensional column vectors and $y_l \in \{1, 2, 3, 4, ..., k\}$ corresponds to class label in training data set and k is number of classes, the vector of ones of appropriate dimensions is denoted by e. c is the penalty parameter and γ is the kernel parameter of the selected kernel. $w_j^t X + b_j = 0$ indicates hyperplane in the feature space of X where w and b are parameters of hyperplane. $m^{(j)}$ denotes the number of patterns in j-th class and n is number of features of each sample. $A^{(j)} \in R^{(m^{(j)} \times n)}$ is the feature matrix of j-th class & $B^{(j)} \in R^{(m-m^{(j)}) \times n}, \forall j = 1, 2, 3, ...j-1, j+1, ..., k$ is the feature matrix of all the classes except j-th class. Transpose of a matrix A is denoted by A^t. i-th row of $A^{(j)}$, represented by $A^{(j)}_{i,.} \in R^n$, is i-th sample of j-th class, having n features. Any x_l is i-th sample of y_l-th class $A^{(y_l)}_{i,.} = x^t_l$. Whole data input is divided into k number of classes each class contains samples of a particular family i.e. and the complete dataset with all the classes is denoted by C. In particular-

$$A^{(j)} = \begin{bmatrix} A^{(j)}_{1,.} \\ A^{(j)}_{2,.} \\ . \\ . \\ . \\ A^{(j)}_{m^{(j)},.} \end{bmatrix}, B^{(j)} = \begin{bmatrix} A^{(1)} \\ . \\ . \\ A^{(j-1)} \\ A^{(j+1)} \\ . \\ . \\ A^{(k)} \end{bmatrix}, C = \begin{bmatrix} A^{(1)} \\ A^{(2)}. \\ . \\ . \\ A^{(k)} \end{bmatrix} \tag{15}$$

In the article, the notation of $A^{(j)}$ represents matrix and (or) set of input vectors of j-th class interchangeably and any new input vector is represented by column vector $X \in R^n$.

Algorithm 1 PNN.

1. X is an input vector
2. For each $A^{(j)}_{i,.}$ in dataset calculate gaussian kernel

$$W_{i,j} = \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp(-\frac{1}{2\sigma^2} \|X - A^{(j)^t}_{i,.}\|^2) \tag{16}$$

where $W_{(i,j)}$ is Gaussian kernel of each sample in the dataset. Sigma (σ) is smoothing parameter. $A^{(j)}_{i,.}$ is i-th sample of j-th class. X is new input feature vector. n indicates length of input vector

3. For each class j calculate the class conditional probability

$$P_j(X) = \frac{1}{m^{(j)}} \sum_{i=1}^{i=m^{(j)}} W_{i,j} \tag{17}$$

where, P_j is the class conditional probability of the j-th class. $m^{(j)}$ is cardinality of j-th class. $W_{i,j}$ is multivariate normal Gaussian kernel of X with i-th training sample of j-th class.

4. Assign the class with highest class conditional probability as predicted class to the new input vector as

$$class(X) = \underset{j}{arg} \max_{1 \le j \le k} P_j(X) \tag{18}$$

2.4.2 Probabilistic neural network

Probabilistic Neural Network (PNN) is Artificial Neural Network (ANN) where sigmoid function is replaced with radial basis function to scale the distance between two vectors. It is statistical approach, based on Parzen Window and Bayesian Network to compute Probability

Density Function (PDF) of each class samples and then Bayes rule is applied to find posterior probability for new input. A new input should be classified to one of the given class which has maximum posterior probability, when no apriori knowledge is given about that class.

PNN use Parzen window and non-parametric approximation technique to find probability density function of jth class. Parzen window specify how to weight all sample in n dimension to determine $P_j(X) = P(X|A^{(j)})$. Basically, training of PNN is prerequisite to find optimized value of smoothing parameter (sigma), that lies between 0 and 1. It is required that training data set must represent actual population thoroughly.

2.4.3 Support vector machine

Support Vector Machine, proposed by Cortes and Vapnik [8], is supervised machine learning algorithm. SVM is a binary classifier, based on the idea of obtaining a maximum margin classification function to adjust the capacity of the model to the size and complexity of the training data. It generates unique hyperplane to separate data point of different classes rather than local boundaries. For the given training dataset T SVM constructs decision function $f(X)$ as-

$$f(X) = sgn(w^t X + b) \tag{19}$$

By finding the separating hyperplane $w.x + b = 0$, where w is a normal vector to hyperplane and b is bias term. To find the value of w and b, primal problem of SVM is defined as below given QP problem.

$$\begin{aligned} \min_{w,b,q} \tfrac{1}{2} \| w \|^2 + ce^t q \\ \text{subject to: } y_i(w^t x_i + b) \geq 1 - q \\ q \geq 0 \\ \forall i = 1, 2, 3, ...m \end{aligned} \tag{20}$$

where $c > 0$ is a regularization parameter.

2.4.4 Twin Support Vector Machine(TWSVM)

Suppose there are only two classes, i.e. (x_i, y_i) is i-th data point, where input $x_i \in R^n$ is given pattern and $y_i \in \{+1, -1\}$ is label for each x_i data point in given training dataset. Let m_1 is data point in class 1 and m_2 is data point in class 2 i.e. $A \in R^{m_1 \times n}$ and $B \in R^{m_2 \times n}$. The goal of TWSVM is to find two non-parallel hyperplanes in n-dimensional input space

$$w_1^t x + b_1 = 0 \text{ and } w_2^t x + b_2 = 0 \tag{21}$$

Such that one hyperplane is closer to the pattern of one class and at least some distance away from the patterns of second class. All the parameters of each hyperplane of TWSVM is obtained by solving a QP problem. The primal problem of TWSVM can be presented as follows.

TWSVM(1)

$$\begin{aligned} \underset{w_1,b_1,q_2}{min} \ \tfrac{1}{2} \| Aw_1 + e_1 b_1 \|^2 + c_1 e_2^t q_2 \\ \text{subject to: } -(Bw_1 + e_2 b_1) + q_2 \geq e_2 \\ q_2 \geq 0 \end{aligned} \tag{22}$$

TWSVM(2)

$$\begin{aligned} &\min_{w_2,b_2,q_1} \frac{1}{2}\|\ Bw_2 + e_2b_2\ \|^2 + c_2e_1^tq_1 \\ &\text{subject to: } (Aw_2 + e_2b_2) + q_1 \geq e_1 \\ &\qquad\qquad\qquad\qquad\qquad q_1 \geq 0 \end{aligned} \tag{23}$$

where the matrix A consists of pattern from positive class and B is the collection of negative class patterns. Positive class contains m_1 point, negative class contain m_2 points and n is number of features in any pattern X. e_1, e_2 are vectors of ones of appropriate size and $c_1, c_2 \geq 0$ are regularization parameters. In TWSVM (1), the objective function of QPP tries to keep hyperplane 1 near to class 1 (positive class) and constraints the negative class data points to be at least at unit distance away from the hyperplane 1. According to Khemchandani and Chandra [14] TWSVM is four time faster than traditional SVM as it solves two small size problem that is approximately of size $(m/2)$.

2.4.5 Multi birth support vector machine(MBSVM)

For multi class classification, Yang et al. [32] proposed MBSVM that is an extension of TWSVM (2) based on decomposition and reconstruction strategy. Proposed classifier solves k-QPP simultaneously, one for each class and predict class label corresponding to new data point X such that given point is farthest from the obtained plane. The hyperplane is obtained by keeping it near to negative class data points and at least one unit away from positive class data points. Consider the j-th class and MBSVM for j-th class can be stated as

MBSVM j-th

$$\begin{aligned} &\min_{w_j,b_j,q_j} \frac{1}{2}\|\ B^{(j)}w_j + e_B^{(j)}b_j\ \|^2 + c_je_A^{(j)^t}q_j \\ &\text{subject to: } \left(A^{(j)}w_j + e_A^{(j)}b_j\right) + q_j \geq e_A^{(j)} \\ &\qquad\qquad\qquad\qquad\qquad q_j \geq 0 \end{aligned} \tag{24}$$

where the matrix $A^{(j)} \in R^{m^{(j)}\times n}$ contains the data point from j-th data class and $B^{(j)} \in R^{(m-m^{(j)})\times n}$ consists of data point from all class except j-th class i.e.

$$\begin{aligned} B_j = &\left[A^{(1)^t}, ..., A^{(j-1)^t}, A^{(j+1)^t}, ...A^{(k)^t}\right] \\ &\qquad\qquad \forall j = 1, 2, 3, ...k \end{aligned} \tag{25}$$

$e_A^{(j)} \in R^{m^{(j)}\times 1}, e_B^{(j)} \in R^{(m-m^{(j)})\times 1}$ are vector of ones of appropriate size, c_j and q_j are penalty parameters and slack variable respectively. w_j and b_j are parameter of j-th hyperplane.

$$w_j^tX + b_j = 0 \tag{26}$$

The decision function of MBSVM is defined as

$$\underset{j}{arg} \max_{1\leq j\leq k} \frac{|w_j^tX + b_j|}{\|\ w_j\ \|} \tag{27}$$

where |.| represent absolute value.

Non linear MBSVM For larger value of k, it is difficult to imagine that "$k-1$ class pattern is proximal to linear hyperplane". To overcome this situation, we need to apply kernel trick to bring data points from original space to higher dimensional space that leads to nonlinear hyperplane. Now in this case decision function is defined as-

$$\begin{aligned} & K(X, C^t)w_j + b_j = 0 \\ & \forall j = 1, 2, 3, ..., k \end{aligned} \tag{28}$$

where X is new data point. $K(X, C^t)$ is a vector generated by appropriately chosen kernel where, C can be defined as

$$C = \left[A^{(1)^t}, ..., A^{(j-1)^t}, A^{(j)^t}, ... A^{(k)^t} \right]^t$$

KMBSVM for j-th class can be stated as

$$\begin{aligned} & \underset{w_j, b_j, q_j}{min} \ \tfrac{1}{2} || \ K(B^{(j)^t}, C^t)w_j + e_B^{(j)} b_j \ ||^2 + c_j e_A^{(j)^t} q_j \\ & \text{subject to: } \left(K(A^{(j)^t}, C^t)w_j + e_A^{(j)} b_j \right) + q_j \geq e_A^{(j)} \\ & q_j \geq 0 \end{aligned} \tag{29}$$

Dual of KMBSVM (29) is defined as

$$\begin{aligned} \underset{\alpha_j}{max} \quad & e_A^{(j)^t} \alpha_j - \tfrac{1}{2} {\alpha_j}^t R^{(j)} \left(S^{(j)^t} S^{(j)} \right)^{-1} R^{(j)^t} \alpha_j \\ & \text{subject to: } 0 \leq \alpha_j \leq c_j \end{aligned} \tag{30}$$

$$R^{(j)} = \left[\begin{array}{cc} K(A^{(j)^t}, C^t) & e_A{}^{(j)} \end{array} \right] \qquad S^{(j)} = \left[\begin{array}{cc} K(B^{(j)^t}, C^t) & e_B{}^{(j)} \end{array} \right]$$

$$Z_j = \left[\begin{array}{cc} w_J & b_j \end{array} \right] = \left(S^{(j)^t} S^{(j)} \right)^{-1} R^{(j)^t} \alpha_j \tag{31}$$

Tikhonov regularization term (ϵI) is introduced to avoid ill conditioned matrix problem in inverse of matrix $\left(S^{(j)^t} S^{(j)} \right)$ and obtained

$$\begin{aligned} \underset{\alpha_j}{max} \ \ & e_A^{(j)^t} \alpha_j - \tfrac{1}{2} {\alpha_j}^t R^{(j)} \left(S^{(j)^t} S^{(j)} + \epsilon I \right)^{-1} R^{(j)^t} \alpha_j \\ & \text{subject to: } 0 \leq \alpha_j \leq c_j \end{aligned} \tag{32}$$

Corresponding decision function is represented as

$$f(x) = \underset{j}{arg} \max_{1 \leq j \leq k} \frac{|K\left(X, E^t\right) w_j + b_j|}{\sqrt{w_j K\left(E^t, E^t\right) w_j}} \tag{33}$$

where |.| represent absolute value.

3 Multi class Twin Support Vector Machine(MTSVM)

Multi birth support vector machine is novel approach for multi class classification problem. It is faster approach, when model is trained with large number of classes as compared to other variants of TWSVM. But in this classification approach, samples from k-th class data points appears in constraints and hence, the hyperplane generated for the class exploits very less information from the k-th class data points. Moreover, this approach generates a hyperplane that could not learn the spatial distribution of the k-th class which is an important aspect of classification. Further, in MBSVM, the samples that are far away from each other, may be classified into the same class and similarity among the same class samples may not be exploited because far away points may be far away in different magnitudes which can be seen in the toy example shown in Figs. 3 and 4.

In order to take the benefits of corresponding class, we propose a novel approach for k class classification with training data set (T). It is also an extension of TWSVM for binary class classification. SImilar to, TWSVM (1) it solves k-QPP and approaches to find a unique hyperplane for each class. Here, hyperplane obtained for j-th class, should be proximal to j-th class samples and at a unit distance from samples of other classes. New data point is assigned a class according to minimum distance from hyperplane.

For given training set T, our linear MTSVM solve k-QPPs simultaneously and find k unique hyperplanes. j-th hyper pane is given by equation as:

$$\begin{aligned} w_j^t x + b_j = 0 \\ \forall j = 1, 2, 3, ..., k \end{aligned} \tag{34}$$

Primal problem of j-th hyperplane is

$$\begin{aligned} \min_{w_j, b_j, q_j} \tfrac{1}{2} || A^{(j)} w_j + e_A^{(j)} b_j ||^2 + c_j e_B^{(j)^t} q_j \\ \text{subject to: } -\left(B^{(j)} w_j + e_B^{(j)} b_j\right) + q_j \geq e_B^{(j)} \\ q_j \geq 0 \end{aligned} \tag{35}$$

where the matrix $A^{(j)}$ is data points from j-th class and $B^{(j)}$ consists of data point from rest of classes. For the above QP Problem constraints require that data pattern of all the classes except j-th class are as far away from j-th hyperplane. Dual of MTSVM is defined as

DMTSVM

$$\begin{aligned} \max_{\alpha_j} \quad e_B^{(j)^t} \alpha_j - \tfrac{1}{2} {\alpha_j}^t G^{(j)} \left(H^{(j)^t} H^{(j)}\right)^{-1} G^{(j)^t} \alpha_j \\ \text{subject to: } 0 \leq \alpha_j \leq c_j \\ H^{(j)} = \left[\, A^{(j)} \;\; e_A{}^{(j)} \,\right] \qquad G^{(j)} = \left[\, B^{(j)} \;\; e_B{}^{(j)} \,\right] \end{aligned} \tag{36}$$

$$U_j = \left(H^{(j)^t} H^{(j)}\right)^{-1} G^{(j)^t} \alpha_j \tag{37}$$

where U_j is a vector of w_j and b_j

$$U_j = \left[\, w_j \;\; b_j \,\right].$$

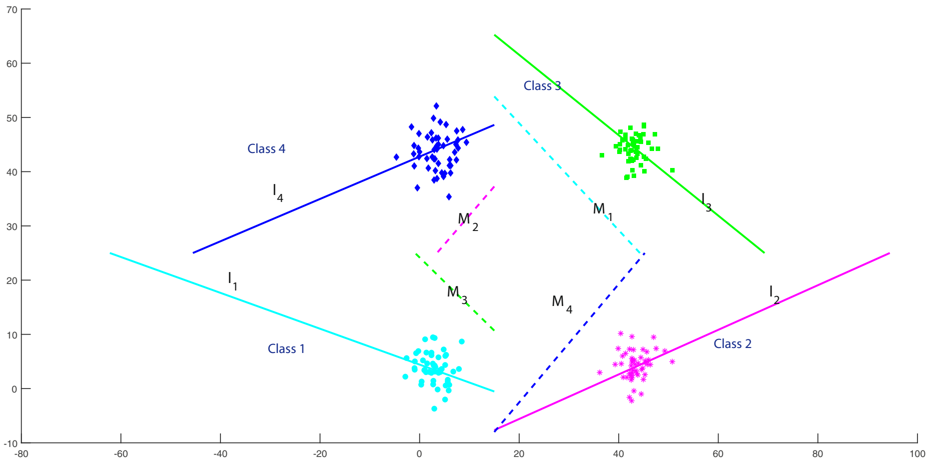

Fig. 3 Toy example representing 4-class classification problem

And the Decision function of MTSVM is given by

$$f(X) = \underset{j}{arg} \min_{1 \le j \le k} \frac{|w_j X + b_j|}{\|w_j\|} \tag{38}$$

To avoid the possibility of ill conditioning of the matrix we introduced the tikhnov's regularisation term ϵI where $\epsilon > 0$ is small scalar entity and I is identity matrix with appropriate dimensions. After introducing the regularisation term in DMTSVM (39) can be defined as

$$\begin{aligned} \max_{\alpha_j} \quad & e_B^{(j)^t} \alpha_j - \tfrac{1}{2} \alpha_j{}^t G^{(j)} \left(H^{(j)^t} H^{(j)} + \epsilon I \right)^{-1} G^{(j)^t} \alpha_j \\ & \text{subject to: } 0 \le \alpha_j \le c_j \end{aligned} \tag{39}$$

$$U_j = \left(H^{(j)^t} H^{(j)} + \epsilon I \right)^{-1} G^{(j)^t} \alpha_j \tag{40}$$

Once the vector U_j is known, we can obtain the hyperplane (34) for each class. Then the new data pattern $X \in R^n$ is assigned to a class as given by (38)

Algorithm 2 Linear MTSVM.

1 For $j = 1$ to k
2 Define $H^{(j)} = \left[A^{(j)} \;\; e_A{}^{(j)} \right]$ $G^{(j)} = \left[B^{(j)} \;\; e_B{}^{(j)} \right]$ where $A^{(j)}$ include samples from j-th class and $B^{(j)}$ represent rest of classes and defined by (25)
3 Select penalty parameter $c_j > 0$
4 Solve (39), (40) and evaluate hyperparameter for plane (34)
5 Assign the class to new data point X by using the decision function (38)

Rational motive behind Multi Class Twin Support Vector Machine (MTSVM) can be represented through two different toy examples. In the first toy example, 4 class classification is considered. These 4 classes are again generated using Gaussian random number generator and mean of all these classes represent the corner of square shape. In Fig. 3, circle, star, square and diamond represents the sample from 4 classes respectively. Separating plane generated through MTSVM i.e. l_1, l_2, l_3 & l_4 and through MBSVM i.e. M_1, M_2, M_3 & M_4 are represented through solid and dashed line respectively. It can be noticed that how a hyperplane generated using MTSVM keeps itself closer to the target class and is away from other classes. On the contrary, MBSVM keeps the sepaerating plane away from its target class and is proximal to other classes.

Next, we considered another toy example with 3 classes such that all of them are centered at co-linear points. All the samples for each class are generated through Gaussian random number generator and mean of all 3 classes are co-linear points in 2-dimensional space. In Fig. 4, circle, diamond and star represent the classes 1, 2 and 3 respectively. These 3 classes are used to train linear MTSVM as well as linear MBSVM. Solid lines l_1, l_2 and l_3 represent the plane generated through MTSVM for 3 classes respectively and dashed lines M_1, M_2 and M_3 represents the plane through MBSVM. As MTSVM learns the spatial information for particular class, it is noted from the Fig. 4 that l_1, l_2 and l_3 are passing through respective class sample while MBSVM generated planes M_1, M_2 and M_3 which are far away from corresponding class and nearer to rest of classes. MBSVM learn very less information for k-th class. From Fig. 4, it can be observed that plane M_2 corresponding to class 2 is farthest plane from class 2 as well as class 3 also. Here, the plane farther away from one class may be farthest away from other classes, that causes higher number of misclassifications.

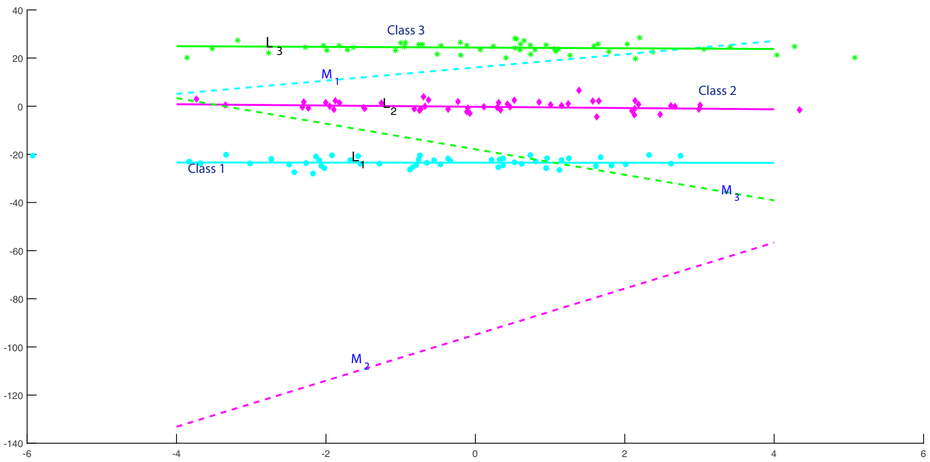

Fig. 4 Toy example representing 3-class classification problem

Non-linear kernel multi class Twin Support Vector Machine Kernel trick is used to achieve non-linear decision surface. Now each hyperplane can be defined as-

$$K(X, E^t)w_j + b_j = 0 \tag{41}$$

where $X \in R^n, E = \left[\begin{array}{cc} A^{(j)^t} & B^{(j)^t} \end{array} \right]$ and $K(x, x)$ is an appropriately chosen kernel. Problem in (35) can be represented using appropriately selected kernel as

$$\begin{aligned} \min_{w_j, b_j, q_j} \ & \tfrac{1}{2}\|\ K(A^{(j)^t}, E^t)w_j + e_A^{(j)} b_j \ \|^2 + c_j e_B^{(j)^t} q_j \\ \text{subject to: } & -\left(K(B^{(j)^t}, E^t)w_j + e_B^{(j)} b_j\right) + q_j \geq e_B^{(j)} \\ & q_j \geq 0 \end{aligned} \tag{42}$$

where the matrix $A^{(j)}$ is data points from j-th class and $B^{(j)}$ consists of data point from rest of classes. Dual of KMCTSVM is given by

DMTSVM j-th

$$\begin{aligned} \max_{\alpha_j} \quad & e_B^{(j)^t} \alpha_j - \tfrac{1}{2} \alpha_j{}^t M^{(j)} \left(N^{(j)^t} N^{(j)}\right)^{-1} M^{(j)^t} \alpha_j \\ & \text{subject to: } 0 \leq \alpha_j \leq c_j \end{aligned} \tag{43}$$

$$M^{(j)} = \left[\begin{array}{cc} K(B^{(j)^t}, E^t) & e_B{}^{(j)} \end{array} \right] \qquad N^{(j)} = \left[\begin{array}{cc} K(A^{(j)^t}, E^t) & e_A{}^{(j)} \end{array} \right]$$

$$Z_j = -\left(N^{(j)^t} N^{(j)}\right)^{-1} M^{(j)^t} \alpha_j \tag{44}$$

Similarly, as in linear classifier after introducing regularization term in inverse of matrix $\left(N^{(j)^t} N^{(j)}\right)$ QP Problem (43) will become-

$$\begin{aligned} \max_{\alpha_j} \quad & e_B^{(j)^t} \alpha_j - \tfrac{1}{2} \alpha_j{}^t M^{(j)} \left(N^{(j)^t} N^{(j)} + \epsilon I\right)^{-1} M^{(j)^t} \alpha_j \\ & \text{subject to: } 0 \leq \alpha_j \leq c_j \end{aligned} \tag{45}$$

$$Z_j = -\left(N^{(j)^t} N^{(j)} + \epsilon I\right)^{-1} M^{(j)^t} \alpha_j \tag{46}$$

After solving QP Problem (45), (46) we obtain the vector $Z_j = \begin{bmatrix} w_j & b_j \end{bmatrix}$. Now we can obtain the hypersurface defined by (41). And the decision function for KMCTSVM is

$$f(x) = \underset{j}{arg} \min_{1 \le j \le k} \frac{|K(X, E^t) w_j + b_j|}{\|w_j\|} \tag{47}$$

Algorithm 3 Non-linear kernel MTSVM.

1 For $j = 1$ to k

2 Define $M^{(j)} = \begin{bmatrix} K(B^{(j)^t}, E^t) & e_B{}^{(j)} \end{bmatrix}$ $N^{(j)} = \begin{bmatrix} K(A^{(j)^t}, E^t) & e_A{}^{(j)} \end{bmatrix}$
where $A^{(j)}$ include samples from j-th class and $B^{(j)}$ represent rest of classes

3 Select penalty parameter $c_j > 0$

4 Solve (45), (46) to find the hyper (41)

5 Assign the class to new data point X by using the decision function (47)

4 Experiments and results

In this section, experimental setup, to analyze the impact of different thresholds, feature sets and classification approaches, is discussed. Further, the results of the experiments are analyzed and discussed.

4.1 Experimental setup

Datasets The efficiency of proposed algorithm is measured using flavia dataset [25]. It is collection of 1907 leaves from 32 common plants in Yangtze Delta, China. Figure 5 shows sample images of each class in flavia dataset. There are 50-70 images of each plant species. Further, each image is 1200 by 1600 pixels in dimensions and stored in jpg file format. We

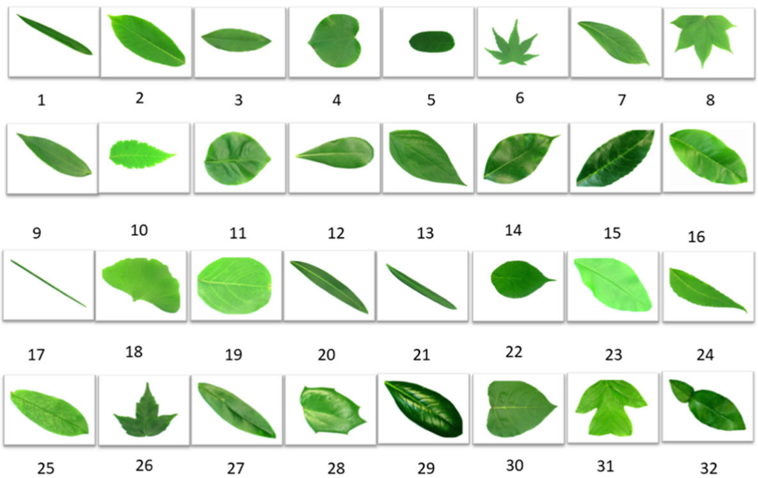

Fig. 5 Sample images from Flavia dataset [25]

have also used another dataset provided by Wu et al. [31] using 1799 leaves from flavia dataset. The only difference between both dataset is threshold value considered for binary conversion. Threshold value of 0.95 is considered by authors in [31], while we used 0.98 as threshold. Table 1, describes the detailed information about the plant species that includes their scientific name, common name and number of plant species in each class.

Optimal hyperparameters A number of experiments are performed to choose best feature set, threshold and classifiers. Performance of these classifiers are based on the

Table 1 Scientific name and common name of plants in Flavia Dataset, number of sample in each class

Class label.	Scientific name	Common name	# sample (Wu et al.)	# sample in Flavia
1	Phyllostachys edulis (Carr.) Houz.	Pubescent bamboo	58	59
2	Aesculus chinensis	Chinese horse chestnut	63	63
3	Berberis anhweiensis Ahrendt	Anhui Barberry	58	65
4	Cercis chinensis	Chinese redbud	72	72
5	Indigofera tinctoria L.	True indigo	72	73
6	Acer Palmatum	Japanese maple	53	56
7	Phoebe nanmu (Oliv.) Gamble	Nanmu	60	62
8	Kalopanax septemlobus Koidz	Castor aralia	51	52
9	Cinnamomum japonicum Sieb.	Chinese cinnamon	51	55
10	Koelreuteria paniculata Laxm.	Goldenrain tree	57	59
11	Ilex macrocarpa Oliv.	Big-fruited Holly	50	50
12	Pittosporum tobira (Thunb.) Ait. f.	Japanese cheesewood	74	63
13	Chimonanthus praecox L.	Wintersweet	38	52
14	Cinnamomum camphora (L.) J.	Presl Camphortree	61	65
15	Viburnum awabuki K.Koch	Japan Arrowwood	58	60
16	Osmanthus fragrans Lour.	Sweet osmanthus	55	56
17	Cedrus deodara (Roxb.) G. Don	Deodar	65	77
18	Ginkgo biloba L.	Ginkgo, Maidenhair tree	57	62
19	Lagerstroemia indica (L.) Pers.	Crape myrtle, Crepe myrtle	57	61
20	Nerium oleander L.	Oleander	61	66
21	Podocarpus macrophyllus (Thunb.)	Sweet yew plum pine	60	60
22	Prunus serrulata Lindl. var. lannesiana auct.	Japanese Flowering Cherry	50	55
23	Ligustrum lucidum Ait. f.	Glossy Privet	52	55
24	Tonna sinensis M. Roem.	Chinese Toon	58	65
25	Prunus persica (L.) Batsch	Peach	50	54
26	Manglietia fordiana Oliv.	Ford Woodlotus	50	52
27	Acer buergerianum Miq.	Trident maple	50	53
28	Mahonia bealei (Fortune) Carr.	Beale's barberry	50	55
29	Magnolia grandiflora L.	Southern magnolia	50	57
30	Populus × canadensis Moench	Canadian poplar	58	64
31	Liriodendron chinense (Hemsl.) Sarg.	Chinese tulip tree	50	53
32	Citrus reticulata Blanco	Tangerine	50	56

Table 2 Cross Validation accuracy of different features set using different fold

Features set	Fold	2	5	10
Shape + Vein	σ	2^{-5}	2^{-4}	2^{-4}
	c	10^{-10}	10^{-10}	10^{-10}
	Accuracy	95.08	97.549	98.11
Shape + HOG	σ	2^{1}	2^{0}	2^{1}
	c	10^{-10}	10^{-10}	10^{-10}
	Accuracy	91.85	95.80	96.30
Shape + LBP	σ	2^{-1}	2^{-1}	2^{-1}
	c	10^{-10}	10^{-10}	10^{-10}
	Accuracy	84.69	92.09	93.56

selection of optimal hyper-parameter. Cross validation is used to select optimal value of hyper-parameters. In this article, we are using PNN, SVM and MBSVM along with the proposed classifier for comparison. In PNN, there is only one parameter sigma (σ) to optimize. Optimal value of sigma is selected from range (0.01, 0.02, ..., 0.09, 0.1, 0.2, ..., 0.9). SVM, MBSVM, MTSVM includes two parameters, penalty parameter (c_j) and kernel parameter gamma (γ) for each decision surface. Since, many decision surfaces has to be constructed for each classifier, same hyper-parameter value is considered for all the decision surfaces for a classifier. The optimal values of penalty parameter c_j and kernel parameter γ are selected by applying grid search cross validation on range range $c = \{10^{-12}, 10^{-10}, ..., 10^{0}, 10^{2}, 10^{4}\}$ and $\gamma = \{2^{-10}, 2^{-9}, ..., 2^{0}, ..., 2^{3}, 2^{4}\}$ respectively. LIBSVM [3] is used to perform experiments of SVM. For multi class classification in SVM, LIBSVM uses one v/s one method. In MBSVM and MTSVM, RBF kernel function is used.

4.2 Comparative analysis of various feature descriptors

Various experiments are performed to analyze different features to find the best features that that can describes leaf images effectively. Here, we extract several features like shape, vein and texture(HOG and LBP), shape feature is concatenated with vein, HOG and LBP. Experiments are performed on three different feature set using proposed classification approach and threshold value 0.98. For better comparative analysis, we have used 2-fold, 5-fold and 10-fold cross validation.

The cross validation accuracy using different features set with optimal value of parameters is shown in Table 2. Table 3 presents the accuracy from previous studies and ten fold

Table 3 Accuracy of Flavia dataset with different feature descriptor

Descriptor	Classification approach	Classification accuracy
Morpholgical,A_{vein}/A_{leaf} [31]	PNN	90.31%
HOG [16]	PNN	84.70%
SMSD,A_{vein}/A_{leaf} [18]	SVM(KNN)	94.50% (78.00%)
SIFT [12]	SVM	95.47%
Shape, Statstical,Vein& fourier descriptor [23]	SVM	88.70%
Automated feature [26]	ResNet26	99.65%
Shape+Vein (Proposed)	MTSVM	98.11%

cross validation accuracy of the proposed classifier. From the results, Shape + Vein feature set is found to be consistently better than all other feature sets. Accuracy in reference [26] is higher than proposed method because technique proposed in [26] is based on deep learning method. There is some limitation with deep learning method as it requires high computation resources as well as large amount of inputs to learn the model. Model computation complexity is also high for deep learning methods. But machine leaning approaches can be used with small dataset with at par accuracy.

4.3 Significance of threshold and classifiers

In this section, we have focused on the experiments that confirms the significance of the threshold value in feature extraction and how it impacts the classification. For comparative analysis, four different classifiers using two different datasets of shape and vein features (i) dataset provided by [31] with threshold value 0.95 and (ii) features extracted using threshold value 0.98 is used. Dataset is divided in such a way that training to testing ratio is same for all k classes. Sub-sampling was performed according to seven different training to testing ratio (RTT){$60 : 40, 70 : 30, 75 : 25, 80 : 20, 85 : 15, 90 : 10, 95 : 5$} and one sample is designed using leave one out sub-sampling method by taking one sample from each class for testing dataset and rest for training dataset. Further, five-fold cross validation technique is used to select the optimal parameter using training dataset. Highest cross validation accuracy is listed in Tables 4 and 5 with optimal parameter using both datasets. From Tables 4 and 5, it is also clear that proposed approach is stable approach because of penalty parameter value is same for all the samples. Further, these optimal parameters are used for training of model using respective training dataset and classification is performed using testing dataset. To test the model MBSVM, MCTSVM both utilizes one v/s other approach. Hence only k-decision surfaces are required. Trained model is tested with unseen data with optimal parameters. The classification accuracy of model is calculated as:

$$Accuracy = \frac{\#correctlyclassifiedsample}{\#testsample} * 100 \tag{48}$$

Table 4 Predictive Accuracy of different classification approach with optimum hyper parameter using Flavia leaf image dataset (1907 leaves) using shape and vein features

Method	RTT	60:40	70:30	75:25	80:20	85:15	90:10	95:5	Leave on out
PNN	σ	0.07	0.08	0.07	0.09	0.08	0.04	0.05	0.08
	Accuracy	76.80	78.27	76.76	77.03	78.66	79.18	78.85	78.95
SVM	c	10^4	10^2	10^4	10^4	10^4	10^4	10^4	10^4
	γ	2^{-6}	2^{-1}	2^{-7}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}
	Accuracy	85.49	85.99	86.13	86.44	86.24	87.28	86.17	86.67
MBSVM	c	10^{-4}	10^{-4}	10^{-2}	10^{-2}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
	γ	2^2	2^2	2	2	2^3	2	2^3	2
	Accuracy	69.798	72.728	71.508	73.559	72.245	74.373	73.623	73.889
MTSVM	c	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}
	γ	2^{-3}	2^{-4}	2^{-4}	2^{-4}	2^{-4}	2^{-3}	2^{-4}	2^{-4}
	Accuracy	94.471	95.803	96.999	96.503	96.082	96.468	97.625	97.497

Table 5 Predictive Accuracy of different classification approach with optimum hyperparameter using dataset given by Wu et al.based on shape and vein features

Method	RTT	60:40	70:30	75:25	80:20	85:15	90:10	95:5	Leave on out
PNN	σ	0.07	0.1	0.09	0.09	0.1	0.1	0.1	0.1
	Accuracy	76.55	75.03	77.43	78.25	77.93	78.61	78.60	78.61
SVM	c	10^4	10^4	10^2	10^4	10^4	10^2	10^4	10^4
	γ	2^{-8}	2^{-6}	2^{-2}	2^{-8}	2^{-9}	2^{-2}	2^{-9}	2^{-6}
	Accuracy	84.22	84.22	84.33	85.20	84.88	84.67	84.60	85.09
MBSVM	c	10^{-2}	10^{-4}	10^{-10}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
	γ	2^0	2^0	2^{-4}	2^0	2^0	2^0	2^0	2^0
	Accuracy	76.87	77.00	76.07	77.11	77.68	77.42	78.71	78.53
MTSVM	c	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^4	10^4
	γ	2^{-3}	2^{-4}	2^{-3}	2^{-2}	2^{-1}	2^{-4}	2^{-1}	2^{-1}
	Accuracy	86.10	88.42	88.60	89.58	89.59	88.98	92.92	93.29

Figure 6, represents the performance of proposed approach and other existing classifier in terms of accurately classified leaf images using unseen test data with thresholds 0.98 and 0.95.

4.4 Analysis of result

Clearly from Table 2, Shape +Vein features outperformed Shape + HOG and Shape+ LBP. It seems reasonable as vein features are more specific to leaves. And hence captures more relevant features from leaf images than others. Wu et al. [31] extracted same features and PCA was used to reduce dimensionality of the feature set. We used the same dataset given and tested with given classifier but it does not produce satisfactory results. Also, feature extraction takes more time because of human intervention and may produce biased result because physiological length is calculated using two points entered by user and results may vary person to person. Further, on application of threshold as stated in [31] some of the leaf images turned black resulting into no feature extraction. And even the dataset provided by them has

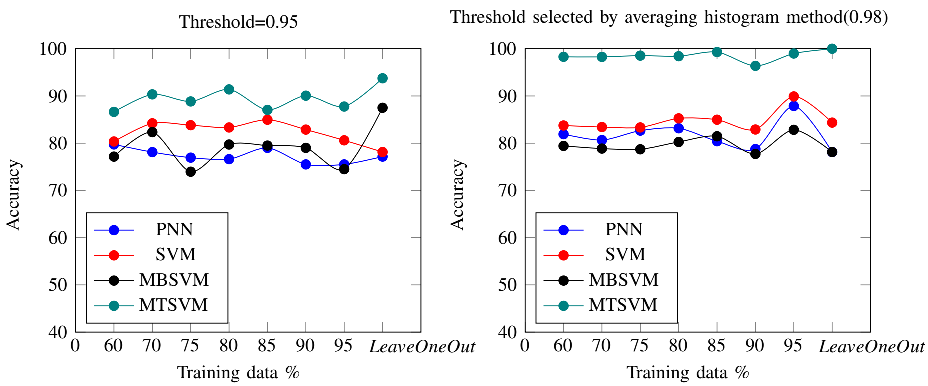

Fig. 6 Performance graph of various classification approaches using optimized parameter with two different threshold value

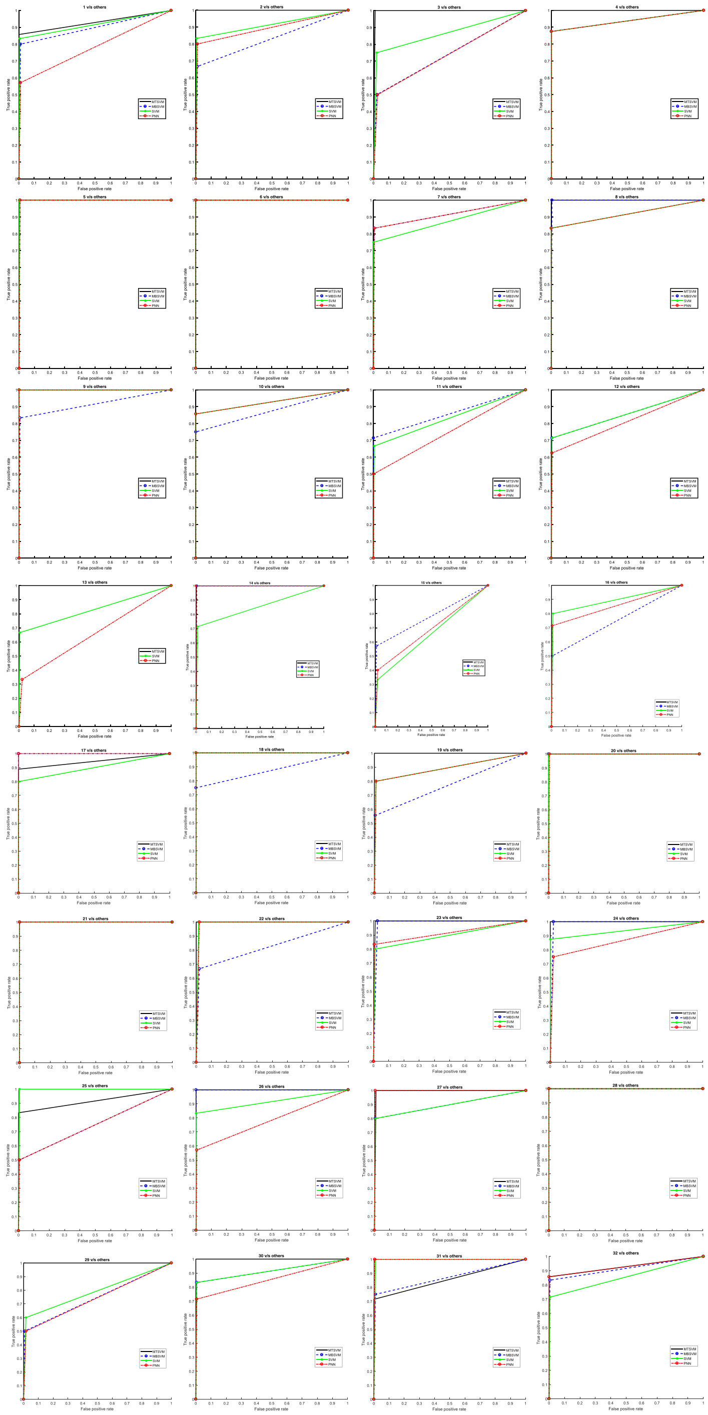

Fig. 7 ROC curve for individual class of FLavia dataset over classification method (threshold=0.98 and 90% training data)

Table 6 Perfromance comparison of classification method over Flavia Dataset

Class	MTSVM	MBSVM	SVM	PNN
1	0.92857	0.89468	0.91399	0.78034
2	1	0.82799	0.91399	0.89468
3	0.99468	0.7393	0.86442	0.73677
4	1	0.9375	0.9375	0.9375
5	1	0.99733	1	0.99733
6	1	1	1	1
7	1	0.91399	0.875	0.91399
8	1	0.99735	0.91667	0.91667
9	1	0.91399	1	1
10	1	0.875	0.92857	0.92857
11	1	0.85714	0.83066	0.75
12	1	0.85445	0.85445	0.8098
13	1	0	0.83066	0.65614
14	1	0.99733	0.85177	1
15	1	0.78034	0.65597	0.68936
16	1	0.74727	0.89468	0.85445
17	0.94444	1	0.9	1
18	1	0.875	1	1
19	1	0.77506	0.89468	0.89468
20	1	1	1	0.99733
21	1	1	1	1
22	1	0.82281	0.99211	0.98953
23	0.99471	0.98698	0.89468	0.91399
24	1	0.98947	0.9375	0.86442
25	0.91667	0.7473	0.99735	0.7473
26	1	1	0.91667	0.78303
27	0.99474	0.89734	0.89734	0.99735
28	1	1	1	1
29	1	0.74727	0.79202	0.74198
30	1	0.91399	0.91399	0.85445
31	0.85714	0.86971	0.99474	0.99735
32	0.92857	0.91399	0.85445	0.92857

features corresponding to 1799 out of 1907 leaf images which again gives proposed threshold an advantage of being applicable to more leaf images. Figure 6 represents the accuracy on Shape +vein features provided by [31] and extracted by keeping threshold 0.98 with different classification techniques with different testing data size using given dataset. Accuracy of PNN at given dataset [31] is in range (75-80)%, while proposed approach accuracy is in range of (87-94)% on the same datset. We extracted same features using regionprops property in Matlab and perform the same experiment 1907 leaves on flavia data [25] and PNN is giving atleast 78.13% using leave one out and atmost 88% accuracy on 95% training data. The proposed approach is giving atleast 97% accuracy. Using leave one out proposed classifier gives 100% accuracy. Results are also compared with SVM and MBSVM. SVM is

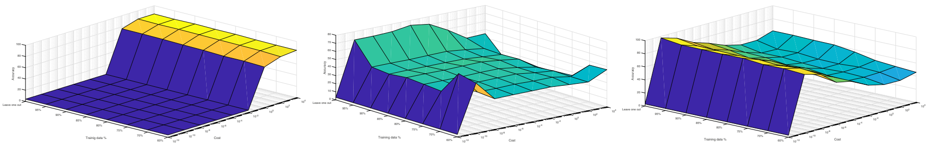

Fig. 8 Performance of classification approaches taking sigma value constant using varying training testing ratio and varying penalty parameter c SVM with constant $\gamma = 2^{-5}$ (left) MBSVM with $\gamma = 2$ (Middle) and Proposed approch MTSVM with $\gamma = 2^{-4}$ (Right)

performing better than PNN and MBSVM for all sample using flavia dataset. This is possibly because MBSVM tries to learn spatial distribution of other classes while being furthest away from the class, to be identified. On the other hand, SVM finds the maximum margin classifier between two classes at a time. This helps SVM find the classifier surface that bears least risk. With both datasets, our proposed approach ourperforms other popular classifiers viz. PNN, SVM and MBSVM. In Tables 4 and 5 cross validation accuracies with optimal parameters is given. These parameters are used in training of classification model along with testing dataset. From Table 4, we can say that proposed model is stable as there in no change in trade off parameter when we change the size of training and testing dataset.

To gain further insight of performance measure of proposed classification method and state of art technique used, ROC curve is plotted for individual classes using 1 v/s others (Fig. 7). The Area under curve is listed in Table 6 for numerical comparison of ROC. The larger the AUC, define the better classification performance of classification method. From the Table 6, it can be noted that for most of classes, AUC for MTSVM is 1, that indicates proposed approaches outperformed other existing method. It is also inferred that for *class 13*, AUC for MBSVM is zero, that indicate, MBSVM fails to learn that particular class and sample from 13 class is misclassified as other classes. For *class 5, 6, 20, 21 and 28* all approches are performing equivalent.

Figure 8, represents the performance of classifier with constant sigma value on different training data and cost parameter. The results confirms that learning the spatial behaviour of the class to be identified (as in MTSVM) is better over finding least risky classifier (as in SVM) and learning other classes spatial distribution.

5 Conclusion and future work

This article discusses MTSVM, a novel version of TWSVM for plant classification based on leaf images. MTSVM, Proposed approach is compared with existing approaches viz. PNN, SVM and MBSVM, results confirmed that proposed approach is robust and effective method. Effectiveness of propsed algorithm is confirmed by performing several experiments on flavia dataset and dataset provided in [31]. Firstly, leaf images are pre-processed for better morphological analysis and then characterized using leaf shape-based features along with vein feature. PCA is used to reduce the dimensionality to solve the problem of high dimensions. Several experiments are performed to compare the efficiency of proposed algorithm and to optimize the hyper-parameter using different training testing ratio. Results shows that MTSVM outperforms the existing method. It is further noticed that the proposed classifier' regularization parameter is very stable over different training sizes. Although, computational time complexity of proposed algorithm is more than existing classifier that lessen the strength of classification system but accuracy wise it shows tremendous improvement. In future, MTSVM can be extended to semi supervised learning approach to utilize the huge amount of unlabelled data.

Acknowledgements We acknowledge University Grant Commission, India for supporting this research by providing fellowship to one of the author, Ms. Neha Goyal. We are also thankful to the reviewers for their valuable and constructive comments and suggestions for the paper. Their inputs have helped us in strengthening the overall quality of the paper.

References

1. Aakif A, Khan MF (2015) Automatic classification of plants based on their leaves. Biosyst Eng 139:66–75
2. Abdi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2(4):433–459
3. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27
4. Chen S, Liu C (2012) Various discriminatory features for eye detection. In: Cross disciplinary biometric systems. Springer, Berlin, pp 183–203
5. Chen S, Liu C (2014) Clustering-based discriminant analysis for eye detection. IEEE Trans Image Process 23(4):1629–1638
6. Chen S, Liu C (2015) Eye detection using discriminatory Haar features and a new efficient SVM. Image Vis Comput 33:68–77
7. Cope JS, Corney D, Clark JY, Remagnino P, Wilkin P (2012) Plant species identification using digital morphometrics: a review. Expert Syst Appl 39(8):7562–7573
8. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
9. Dalal N, Triggs B, Schmid C (May) Human detection using oriented histograms of flow and appearance. In: European conference on computer vision. Springer, Berlin, pp 428–441
10. Dallimer M, Irvine KN, Skinner AM, Davies ZG, Rouquette JR, Maltby LL, Gaston KJ (2012) Biodiversity and the feel-good factor: understanding associations between self-reported human well-being and species richness. BioScience 62(1):47–55
11. Du JX, Wang XF, Zhang GJ (2007) Leaf shape based plant species recognition. Appl Math Comput 185(2):883–893
12. Hsiao JK, Kang LW, Chang CL, Lin CY (2014) Comparative study of leaf image recognition with a novel learning-based approach. In: Science and information conference (SAI), 2014. IEEE, pp 389–393
13. Kalyoncu C, Toygar O (2015) Geometric leaf classification. Comput Vis Image Underst 133:102–109
14. Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
15. Kumar N, Belhumeur PN, Biswas A, Jacobs DW, Kress WJ, Lopez IC, Soares JV (2012) Leafsnap: a computer vision system for automatic plant species identification. In: Computer vision–ECCV 2012. Springer, Berlin, pp 502–516
16. Pham NH, Le TL, Grard P, Nguyen VN (2013) Computer aided plant identification system. In: 2013 International conference on computing, management and telecommunications (ComManTel). IEEE, pp 134–139
17. Pilgrim SE, Cullen LC, Smith DJ, Pretty J (2008) Ecological knowledge is lost in wealthier communities and countries
18. Priya CA, Balasaravanan T, Thanamani AS (2012) An efficient leaf recognition algorithm for plant classification using support vector machine. In: 2012 International conference on pattern recognition, informatics and medical engineering (PRIME). IEEE, pp 428–432
19. Punyasena SW, Smith SY (2014) Bioinformatic and biometric methods in plant morphology. Appl Plant Sci 2(8):1400071
20. Robinson BS, Inger R, Gaston KJ (2016) A rose by any other name: plant identification knowledge & socio-demographics. PloS one 11(5):e0156572
21. Rosenfeld A, De La Torre P (1983) Histogram concavity analysis as an aid in threshold selection. IEEE Trans Syst Man Cybern 2:231–235

22. Rzanny M, Seeland M, Wäldchen J, Mäder P (2017) Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain. Plant methods 13(1):97
23. Saleem G, Akhtar M, Ahmed N, Qureshi WS (2019) Automated analysis of visual leaf shape features for plant classification. Comput Electron Agric 157:270–280
24. Sezan MI (1990) A peak detection algorithm and its application to histogram-based image data reduction. Comput Vis Graph Image Process 49(1):36–51
25. Sourceforge (2017) Flavia: a leaf recognition algorithm for plant classification using pnn. http://flavia.sourceforge.net/
26. Sun Y, Liu Y, Wang G, Zhang H (2017) Deep learning for plant identification in natural environment. Comput Intell Neurosci, 2017
27. Tomar D, Agarwal S (2015) A comparison on multi-class classification methods based on least squares twin support vector machine. Knowl-Based Syst 81:131–147
28. Trias-Blasi A, Vorontsova M (2015) Botany: plant identification is key to conservation. Nature 521(7551):161
29. Wäldchen J, Mäder P (2018) Plant species identification using computer vision techniques: a systematic literature review. Archiv Comput Methods Eng 25(2):507–543
30. Wäldchen J, Rzanny M, Seeland M, Mäder P (2018) Automated plant species identification—trends and future directions. PLoS Comput Biol 14(4):e1005993
31. Wu SG, Bao FS, Xu EY, Wang YX, Chang YF, Xiang QL (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. In: 2007 IEEE International symposium on signal processing and information technology. IEEE, pp 11–16
32. Yang ZX, Shao YH, Zhang XS (2013) Multiple birth support vector machine for multi-class classification. Neural Comput Applic 22(1):153–161
33. Yanikoglu B, Aptoula E, Tirkaz C (2014) Automatic plant identification from photographs. Mach Vis Appl 25(6):1369–1383

Neha Goyal graduated and Masters in Computer Science from Kurukshetra University, Kurukshetra, India. She is awarded by Gold Medal in M.Sc.(Computer Science). Currently, she is pursuing Ph.D. from National Institute of Technology, Kurukshetra. Her research interests include Image Processing and Machine learning.

Dr. Kapil Gupta is currently working as an Assistant Professor at National Institute of Technology, Kurukshetra. He received his Doctor of Philosophy and Master of Technology from Jawaharlal Nehru University, Delhi. His research interests include Machine Learning methods,Wireless Sensor Networks and applications.

Nitin Kumar He has done his Doctor of Philosophy from School of Computer and Systems Sciences, Jawaharlal Nehru University in 2015. He is currently working as Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology Uttarakhand since 2013. His current research areas are Pattern Recognition, Image Processing and Salient Object Detection.