One-Dimensional vs. Two-Dimensional based Features: Plant Identification Approach

Alaa Tharwat[a,b], Tarek Gaber[b,c,1,*], Aboul Ella Hassanien[b,d]

[a] *Faculty of Engineering, Suez Canal University, Egypt*
[b] *Scientific Research Group in Egypt (SRGE), http://www.egyptscience.net*
[c] *Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt*
[d] *Faculty of Computers and Information, Cairo University, Egypt*

Abstract

The number of endangered species has been increased due to shifts in the agricultural production, climate change, and poor urban planning. This lead to investigating new methods to address the problem of plant species identification/classification. In this paper, a plant identification approach using 2D digital leaves images was proposed. The approach used two features extraction methods based on one-dimensional (1D) and two-dimensional (2D) and the Bagging classifier. For the 1D-based methods, Principal Component Analysis (PCA), Direct Linear Discriminant Analysis (DLDA), and PCA+LDA techniques were applied, while 2DPCA and 2DLDA algorithms were used for the 2D-based method. To classify the extracted features in both methods, the Bagging classifier, with the decision tree as a weak learner was used. The five variants, i.e. PCA, PCA+LDA, DLDA, 2DPCA, and 2DLDA, of the approach were tested using the Flavia public dataset which consists of 1907 colored leaves images. The accuracy of these variants was evaluated and the results showed that the 2DPCA and 2DLDA methods were much better than using the PCA, PCA+LDA, and DLDA. Furthermore, it was found that the 2DLDA method was the best one and the increase of the weak learners of the Bagging classifier yielded a better classification accuracy. Also, a comparison with the most related work showed that our approach achieved better accuracy under the same dataset and same experimental setup.

Keywords: Plant Identification, Principal Component Analysis, Linear

Discriminant Analysis (LDA), Bagging classifier, weak learners, 2DLDA, 2DPCA, Direct-LDA, Leaf Image, Leaves Images, Small Sample Size (SSS), PCA+LDA

1. Introduction

Plants are a vital element of the Earth's ecology system. They maintain a healthy breathable atmosphere. Almost the entire oxygen, needed for humans and other animals breathe, are produced by plants, thus without plants, there is no life on the earth (Gaber et al., 2015; Chaki et al., 2016). In addition, plants can be used as an alternative energy source, e.g., bio-fuel (Chaki and Parekh, 2012). There are various species of plants which are subject to the danger of extinction. Saving endangered species of these plants from becoming extinct and protecting their wild places is important for our health and the future of our children. The impact of biodiversity loss may lead to fewer new medicines, greater vulnerability to natural disasters and greater effects from global warming. Therefore, there is a need for protecting plants and classifying them into different species. For this purpose, plant identification techniques have become a hot area of research.

Traditional plant identification can be achieved by a manual matching of the plant's characteristics including leaves, fruits, flowers, and stem, against an atlas. Such identification requires extensive knowledge and it makes use of complex terminology in a way that even a professional botanist needs to spend much time in a field to achieve plant identification. The plant identification could be automatically achieved through using the plants' features that are extracted from their images and then these features can be classified using various classifier techniques such as, Neuro-Fuzzy Classifier (Chaki et al., 2016), Support Vector Machine (SVM) (Arun Priya et al., 2012a), etc. Since some plants' flowers and fruits are seasonal and their colors are changed according to the season, the leaves are more suitable to identify plants than flowers and fruits. Hence, the majority of the existing computer-based plant identification has used the leaves of plants (Chaki and Parekh, 2012; Chaki et al., 2015, 2016). The automatic plant identification based on information technology is a very vital task for different parties: agriculture, pharmacological, forestry science. Automatic plant identification process will achieve fast, cheap, and accurate systems, which provide a great help to medicine, industry, and foodstuff production, as well as to biologists, chemists, and environmentalists.

This paper describes an approach addressing the plant identification problem by using features that are extracted from digital images of plant leaves as it is a low-cost and convenient way to get leaf images dataset. The approach used two features extraction techniques (one-dimensional (1D) and two-dimensional (2D) based) with the Bagging classifier. For the 1D-based techniques, PCA, PCA+LDA, and Direct-LDA techniques were applied, while 2DPCA and 2DLDA algorithms were used for the 2D-based method. To classify the extracted features in both methods, the Bagging classifier, with the decision tree as a weak learner was used.

The rest of the paper is organized as follows; Section (2) summarizes the related work of the plant identification based on machine learning. Section (3) highlights the feature extraction methods and the classifier used in the design of the proposed approach which is presented in Section (4). The experimental results are reported in Section (5) while the results' discussion and the conclusion are presented in Section (6) and Section (7), respectively.

2. Related Works

There are a number of plant identification approaches that used digital images (Valliammal and Geethalakshmi, 2011; Arora et al., 2012; Arun Priya et al., 2012b; Satti et al., 2013). Satti *et al.* classified plant leaves based on 2D images. They used Flavia image dataset and applied many preprocessing steps on the leaf images (Satti et al., 2013). Their approach achieved accuracy 85.9% and 93.3% using k-Nearest Neighbour (k-NN) and Artificial Neural Networks (ANN) classifiers, respectively. Arora *et al.* applied the Speed Up Robust Features (SURF) to extract the features from leaf images and then used the Random Forest (RF) classifier and tested their approach using Plant Leaves II dataset (Arora et al., 2012). In another research, Caglayan *et al.* utilized color and shape features to classify 32 different kinds of plants. They used SVM, k-NN, RF, and Naive Bayes (NB) classifiers and the RF classifier achieved the best accuracy (96%) (Caglayan et al., 2013). Arun *et al.* transformed the leaf images into grayscale and applied boundary enhancement operations (Arun Priya et al., 2012b). They then used the PCA to extract features and then used SVM and k-NN for classification. They used Flavia dataset and achieved the accuracy of 78% to 81.3% using k-NN classifier.

Valliamma *et al.* proposed identification approach for flower images dataset (Valliammal and Geethalakshmi, 2011). They applied Preferential Image Segmentation (PIS) and other enhancement operations to the images. They then used the image thresholding to obtain some features and then used the probabilistic curve for classification. They used a dataset of 500 flowers images. In another research, Uluturk and Uger converted the plant leaf images into grayscale, the region of interest was segmented and the features were extracted (Uluturk and Ugur, 2012). Probabilistic Neural Networks (PNN) classifier was then used of Flavia dataset and the classification rate was 92.5%.

Recently, Chaki *et al.*, proposed a plant recognition approach using both of texture and shape features (Chaki et al., 2015). The texture features were extracted by Gray Level Co-occurrence Matrix (GLCM) and Gabor filter while the shape features were extracted using the curvelet transform coefficients and the invariant moments. This approach was tested using two neural-based classifiers: a feed-forward back-propagation Multi-Layered Perceptron (MLP) and a Neuro-Fuzzy Classifier (NFC) to classify 31 plant species of leaves images. In another study, (Chaki et al., 2016) proposed another approach based on ridge filter and curvelet transform with a Neuro-Fuzzy classifier. The classification accuracy of almost all classes (plant species) was 100%. However, it needs preprocessing step which imposes more CPU time.

3. Preliminaries

In this section, the background of the PCA and LDA methods are introduced. Moreover, the details of how to use both methods in vector or matrix form are explained below.

3.1. Feature Extraction Method

The aim of the feature extraction step is to transform the objects' properties into numeric values. There are many types of features for an image such as shape, texture, and color features. The shape features are used to describe the shape of the image or the Region of Interest (ROI) while the texture features describe the texture analysis of the image. The texture features methods are generally classified into two methods: sparse method and dense method. In the sparse method, the interest points are first detected and then a local patch around these points is constructed, and finally, invariant features are extracted. *Scale Invariant Feature Transformation* (SIFT) is one of the most common algorithms in the sparse descriptor method (Lowe, 1999; Tharwat et al., 2015). In the dense method, the local features are extracted from each pixel over the input image. *Local Binary Patterns* (LBP) is one of the most common algorithms in dense method (Ojala et al., 2002; Tharwat et al., 2014b). The color features are widely used in image retrieval due to its robustness against image size variation and orientation (Salvador et al., 2004). The feature extraction techniques used in the proposed approach are highlighted below.

3.1.1. An Overview of PCA

(PCA) is one of the classical feature extraction techniques that is widely used in the areas of pattern recognition and computer vision since Turk and Pentland (Turk and Pentland, 1991) used it for face recognition in 1991. From that time, PCA has been widely used in face recognition and many other pattern recognition applications such as dimensionality reduction (Moore, 1981), face recognition (Turk and Pentland, 1991; Yang et al., 2004), and ear recognition (Tharwat et al., 2012).

The PCA is an unsupervised method that is used to search for a new space (PCA space or eigen space), W_{PCA}, which reduces the d-dimensional feature vectors to k-dimensional feature vectors (where $k < d$).

Given $I = \{I_1, I_2, \ldots, I_M\}$, where $I_i \in \mathcal{R}^d$ is the i^{th} pattern or sample, d is the dimension or the number of features of I_i, and M is the total number of samples. PCA searches for the PCA space (W_{PCA}) which represents the direction of the maximum variance of the given data. The PCA space consists of k orthonormal and uncorrelated Principal Components (PCs). The first step of the PCA method is to calculate the covariance matrix Σ as follows:

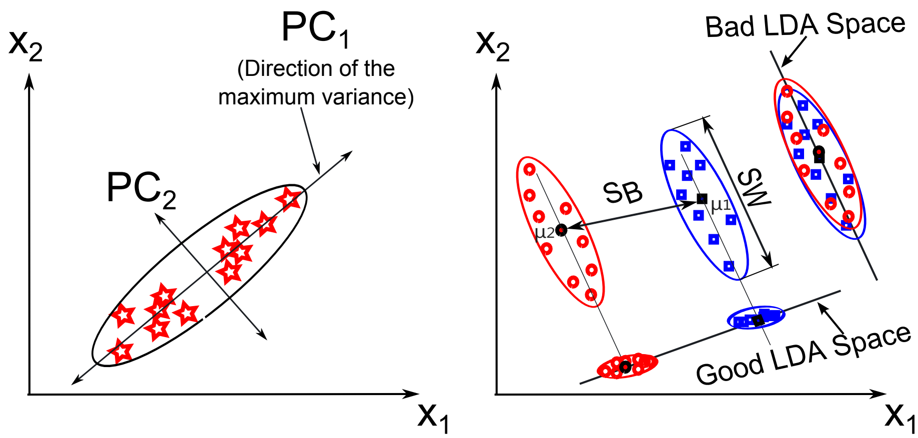
$$\Sigma = \frac{1}{M-1} D \times D^T, \tag{1}$$

$$D = \{d1, d2, \ldots, d_M\} = \sum_{i=1}^{M} I_i - \mu \tag{2}$$

where $\mu = \frac{1}{M}\sum_{i=1}^{M} I_i$ is the mean of all samples. The eigenvalues ($\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$) and eigenvectors ($\{v_1, v_2, \ldots, v_d\}$) of Σ are then calculated. The eigenvector with the highest eigenvalue represents the first principal component and it has the maximum variance as shown in Figure 1a (Turk and Pentland, 1991; Strang, 2003). As shown in the figure, the first principal component (PC1) points to the maximum variance. Algorithm (1) summarizes the steps of the PCA technique.

3.1.2. An Overview of LDA

LDA is also a well-known algorithm for feature extraction and dimensionality reduction. LDA is widely used in different applications such as biometrics (Marcialis and Roli, 2002; Tharwat et al., 2014a), bioinformatics (Wu et al., 2009), and chemoinformatics (Mitchell, 2014). LDA is a supervised dimensionality reduction and feature extraction method (Galdámez et al., 2015). It finds the projection space that maximizes the ratio of the between-class variance, S_B, to the within-class variance, S_W, and hence guaranteeing maximum class separability as shown in Figure 1b (Welling, 2005). From the figure, there are two sub-spaces that can be selected to represent the LDA space. As shown, in



(a) (b)

Figure 1: A visualization of the PCA and LDA techniques; (a) PCA, (b) LDA.

Algorithm 1 : PCA

1: Given a feature matrix which consists of all training samples, each sample is represented by a single column as follows, $I = [I_1, I_2, \ldots, I_M]$, where M represents the total number of samples, I_i represents a training sample.

2: Compute the mean of all classes (total mean) $\mu = \frac{1}{M}\sum_{i=1}^{M} I_i$.

3: Subtract the mean from all training samples as follows, $D_i = I_i - \mu$.

4: Compute covariance matrix $Cov = \frac{1}{M-1}\sum_{i=1}^{M} D_i * D_i^T$.

5: Compute eigenvectors V and eigenvalues λ of the covariance matrix.

6: Sort eigenvectors according to their corresponding eigenvalues.

7: Select k eigenvectors that have the largest eigenvalues $W_{PCA} = \{v_1, v_2, \ldots, v_k\}$. The selected eigenvectors represent the projection space of PCA (W_{PCA}).

the bad LDA space, the two classes cannot be discriminated because the S_B between the two classes decreased. On the other hand, in the good LDA space, S_W is decreased while S_B is increased and hence the two classes are perfectly discriminated.

Assume the training samples belong to C classes. The aim of the LDA method is to search for the subspace, W_{LDA}, which maximizes S_B and minimizes S_W as follows:

$$J(w) = \frac{W_{LDA}^T S_B W_{LDA}}{W_{LDA}^T S_W W_{LDA}}, \tag{3}$$

$$S_B = \sum_{i=1}^{C} \frac{n_i}{M} (\mu_i - \mu)(\mu_i - \mu)^T, \tag{4}$$

$$S_W^i = \frac{1}{n_i} \sum_{j=1}^{n_i} (I_j^i - \mu_i)(I_j^i - \mu_i)^T, \tag{5}$$

$$S_W = \sum_{i=1}^{C} \frac{n_i}{M} S_W^i \tag{6}$$

where n_i is the number of samples of class i, $\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} I_j^i$ is the mean of class i, $\mu = \frac{1}{C} \sum_{i=1}^{C} \mu_i = \frac{1}{M} \sum_{j=1}^{M} I_j^i$ represents the global mean or the mean of all samples, I_j^i is the j^{th} sample in the i^{th} class, $M = \sum_{i=1}^{C} n_i$, and S_W^i is the within-class matrix of the i^{th} class. Algorithm (2) summarizes the steps of the LDA technique.

In practice, S_W is always singular, this is the so-called singularity, Small Sample Size (SSS), or under-sampled problem. This problem is common in LDA technique and it results from high-dimensional pattern classification applications or a small number of training samples available for each class compared with the dimensionality of the sample space (Lu et al., 2005; Ye and Xiong, 2006). The SSS problem occurs when the S_W is singular[2]. The upper bound of the rank[3] of S_W is $M - C$, while the dimension of S_W is $d \times d$ (Lu et al., 2005; Feng and Wu, 2014). Thus, in most cases $d >> M - C$ which leads to SSS problem. For example, in face recognition applications, the size of the face image may reach

[2] A matrix is singular if it is square, does not have a matrix inverse, and/or its determinant is zero; hence not all columns and rows are independent (Strang, 2003).

[3] The rank of the matrix represents the number of linearly independent rows or columns (Strang, 2003).

to $100 \times 100 = 10000$ pixels, which represent high-dimensional features and it leads to a singularity problem.

There are two common solutions to SSS problem. The first solution is to use a non-singular intermediate, e.g. PCA space, to reduce the dimension of the original data to be equal to the rank of S_W, hence S_W becomes full-rank and S_W can be inverted. The second solution is to remove the null-space of S_B which contains no useful information for recognition by diagonalizing S_B and then diagonalizing S_W. These two solutions were used in this paper.

Algorithm 2 : Linear Discriminant Analysis (LDA)

1: Given a set of M samples $[I_i]_{i=1}^{M}$, each of which is represented as a column as follows, $I = [I_1, I_2, \ldots, I_M]$ and each sample is represented by d features.

2: Compute the mean of each class, μ_i, and the total mean of all samples, μ.

3: Compute within-class scatter matrix, S_W, as in Equations (5 and 6) and the between-class scatter matrix S_B as in Equation (4).

4: Calculate the eigenvalues (λ) and eigenvectors (V) of $S_W^{-1} S_B$ as follows:

$$S_B V = S_W V \lambda \tag{7}$$

5: Sort the eigenvectors in descending order according to their corresponding eigenvalues, then use the first, k, eigenvectors as a lower dimensional space (W_{LDA}).

3.1.3. One-Dimensional Feature Extraction Technique:

The classical PCA (i.e. 1DPCA) and LDA (i.e. 1DLDA) use one-dimensional/vector form to calculate projection spaces as shown in Figure 2. In both methods, a two-dimensional image ($I_i(r \times c)$, $\forall\, i = 1, 2, \ldots, M$) is first converted into one feature vector (column or row), where r and c represent the number of rows and columns of the image, respectively. All the feature vectors are then concatenated to form a feature matrix ($I = \{I_1, I_2, \ldots, I_M\}$), where M refers to the total number of images. The PCA and LDA spaces, W_{PCA} and W_{LDA}, of this matrix (I) can be calculated. The features are then extracted by projecting the feature matrix on the calculated spaces as follow, $Y = W^T I$, where W represents the lower dimensional space (i.e. PCA or LDA) (see Figure 3a).

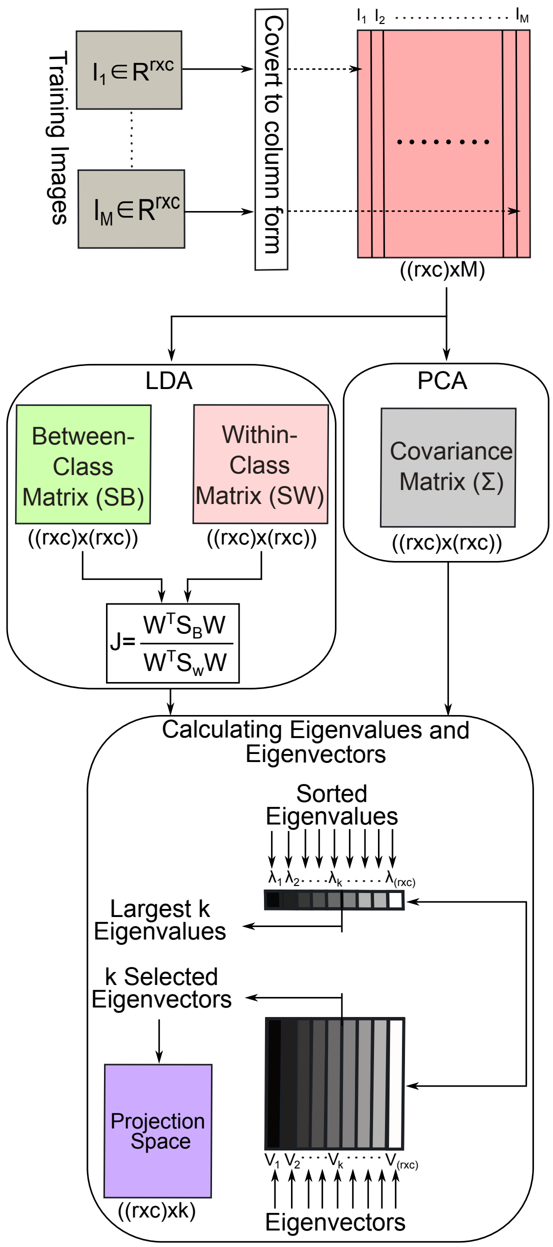
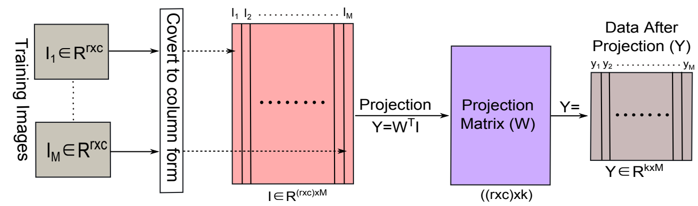
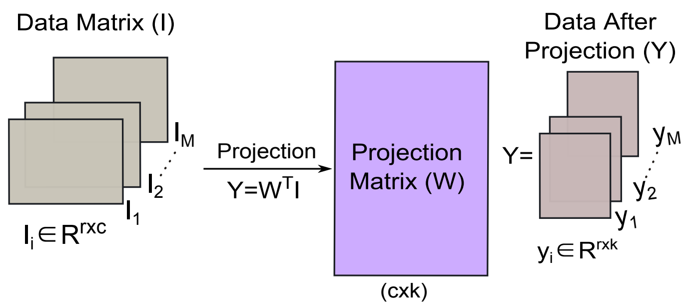


Figure 2: Visualized steps to calculate a projection space of one-dimensional PCA and LDA (1DPCA and 1DLDA) methods.

Vector representation may lead to a high-dimensional data. Hence, it is difficult to calculate the covariance matrix in PCA due to its large size. Moreover, the high-dimensional data leads to SSS problem in LDA. These two problems can be solved using the two-dimensional methods, i.e. 2DPCA and 2DLDA.



(a)



(b)

Figure 3: A visualization of the projection of one-dimensional and two-dimensional methods; (a) one-dimensional method, (b) two-dimensional method.

3.1.4. Two-Dimensional Feature Extraction Techniques

The spaces of the PCA and LDA techniques can be calculated in two-dimensional/matrix form, i.e. 2DPCA and 2DLDA, as shown in Figure 4. Hence, there is no need for the step of converting each image into one vector prior to feature extraction step which saves more computational time. As in one-dimensional technique, the PCA and LDA spaces, W_{PCA} and W_{LDA}, are calculated and the features are then extracted by projecting the feature matrix on the calculated spaces as follows, $Y = W^T I$ (see Figure 3b).

3.1.4.1. Two Dimensional PCA (2DPCA). The aim of the 2DPCA method is to find the PCA space, W_{PCA}, to project the two-dimensional image ($I_i \in \mathcal{R}^{r\times c}$) as follows, $Y_i = W_{PCA}^T I_i$, where Y_i is the projected feature vector of the image I_i. First, the M two-dimensional images are used to calculate the covariance matrix ($\Sigma \in \mathcal{R}^{c\times c}$) as in Equation (8). The eigenvalues and eigenvectors of Σ are then calculated and k optimal eigenvectors, i.e. projection axes, are selected. In other words, the 2DPCA method then searches for the PCA space $W_{PCA} = \{v_1, v_2, \ldots, v_k\}$ which maximizes the variance as in classical PCA, where v_i is the i^{th} principal component and k is the number of selected eigenvectors that represent the PCA space. This projection space is used for feature extraction of the image as follows, $Y_i = W_{PCA}^T I_i$, where $Y_i \in \mathcal{R}^{r\times k}$ represents the projected feature vectors, i.e. feature matrix or feature image, of the image I_i (Yang et al., 2004).

$$\Sigma = \frac{1}{M-1}\sum_{j=1}^{M}(I_j-\mu)^T(I_j-\mu) \tag{8}$$

where μ is the mean of all training images, M is the number of training images, and I_j represents the j^{th} training image.

3.1.4.2. Two Dimensional LDA (2DLDA). The aim of the 2DLDA method is to find the LDA space, W_{LDA}, to extract the features by projecting the two-dimensional image on the LDA space using $Y_i = W_{LDA}^T I_i$. Assume I_i represents one image and M two-dimensional images are used to calculate within-class matrix (S_W) and between-class variance (S_B). The eigenvalues and eigenvectors of $S_W^{-1} S_B$ are then calculated and k optimal eigenvectors are selected to form the LDA space, i.e. Fisher projection matrix using $W_{LDA} = \{v_1, v_2, \ldots, v_k\}$ which maximizes the ratio between S_B and S_W as in classical LDA, where v_i is the i^{th} eigenvector.

3.2. The Bagging Classifier

The Bagging classifier is one of the ensemble classifiers creating its ensemble by training different classifiers or weak learners on a random distribution of

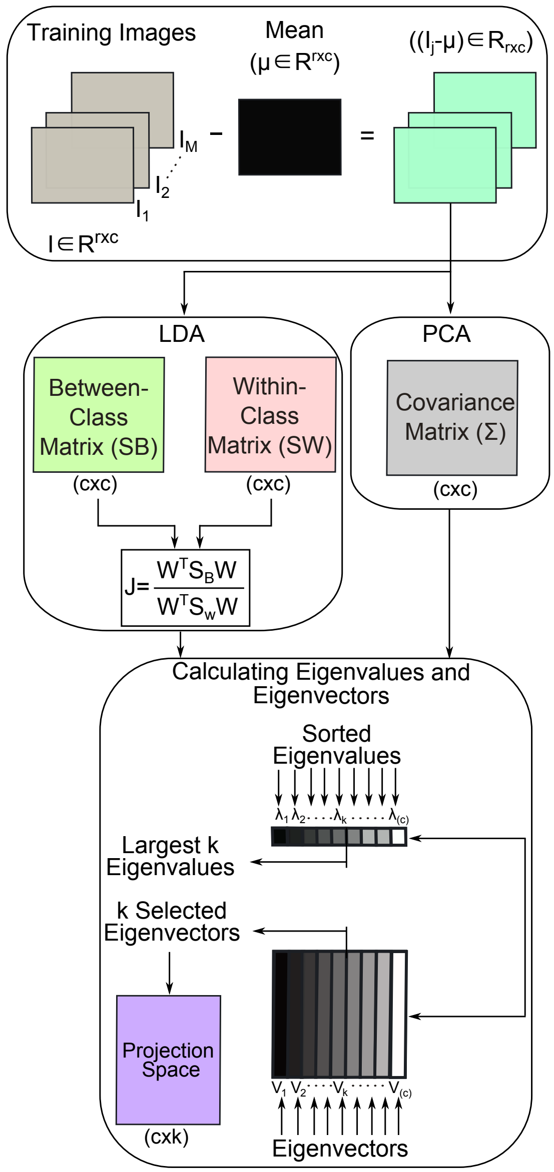


Figure 4: Visualized steps to calculate a projection space of two-dimensional PCA and LDA (2DPCA and 2DLDA) methods.

a training dataset. A weak learner is a simple, fast, and easy to implement classifier such as single level decision tree or simple neural networks (Kuncheva, 2014).

Generally, as given in Algorithm (3), a Bagging classifier consists of two phases: training and testing. In the training phase, for each iteration, t, a number of training samples are selected randomly (S_i), and these samples are used to train one weak learner (C_t) as shown in Figure 5. In the testing phase, all the weak learners are used to classify an unknown sample (I_{test}). The outputs of all weak learners are combined using majority voting method to determine the final decision (Kuncheva, 2014).

Algorithm 3 Bagging Classifier Algorithm

1: Given a training set $I = (I_1, y_1), \ldots, (I_M, y_M)$, where y_i represents the label of samples $I_i \in I$ and M denotes the total number of samples in the training set.
2: **while** $(t < T)$ **do**
3: Select a sample S_t from I.
4: Use S_t to train the current weak learner C_t.
5: **end while**
6: Given new test pattern I_{test}.
7: Classify I_{test} using all weak learners.
8: Combine the outputs of all weak learners to determine the final prediction.

4. Proposed Approaches

The proposed plant identification approach consists of two phases. In the first phase, two main feature extraction methods (1D-based and 2D-based) were used. In the 1D-based feature extraction method, 1DPCA, Direct LDA (DLDA), and (PCA+LDA) techniques were used while in the 2D-based method, 2DPCA and 2DLDA were applied for the feature extraction step. For the identification, in both techniques, the Bagging classifier was used to identify the type of the unknown leave image as shown in Figure 5. As shown in Figure 5, the proposed model has two main phases: training and testing phases.

4.1. Training Phase

In the training phase, M images ($I_{i=1}^{M}$) were used to train the proposed model. In the 1D-based method, each image was first transformed into one vector and then all training images' vectors were combined into a matrix, $I = [I_1, I_2, \dots, I_M]$ (see Figure 2). In the 2D-based method, the training image was not changed but represented as 2D matrix as seen in Figure 4. The PCA or LDA spaces, W, of I were then constructed. The features were then extracted from all training images by projecting the images on the space. These features were used to train the Bagging model. The steps of the training phase are explained in detail in Algorithm (4).

Algorithm 4 : Training Phase

1: Read the training images.
2: **if** (1D-based method) **then**
3:     Convert all images $I_i(r \times c), i = 1, \dots, M$ into vectors $I_i((r \times c) \times 1)$.
4:     Combine all feature vectors into a matrix ($I = [I_1, I_2, \dots, I_M]$).
5: **else**
6:     Deals with images in 2D form (i.e. matrix representation).
7:     Combine all feature vectors into a matrix ($I = [I_1, I_2, \dots, I_M]$).
8: **end if**
9: Compute the projection space (W).
10: Project I on the projection surface (W) to obtain the features as follows, $Y = W^T I$.
11: Train the Bagging classifier using the extracted features, Y.

4.2. Testing Phase

In the testing phase, an unknown leave image (I_{test}) was tested for its plant identification. To do so, firstly the leave features were extracted by projecting it on the projection space, W, that was computed in the training phase, i.e. $Y_{test} = W^T I_{test}$. The computed vector Y_{test} was classified using the Bagging

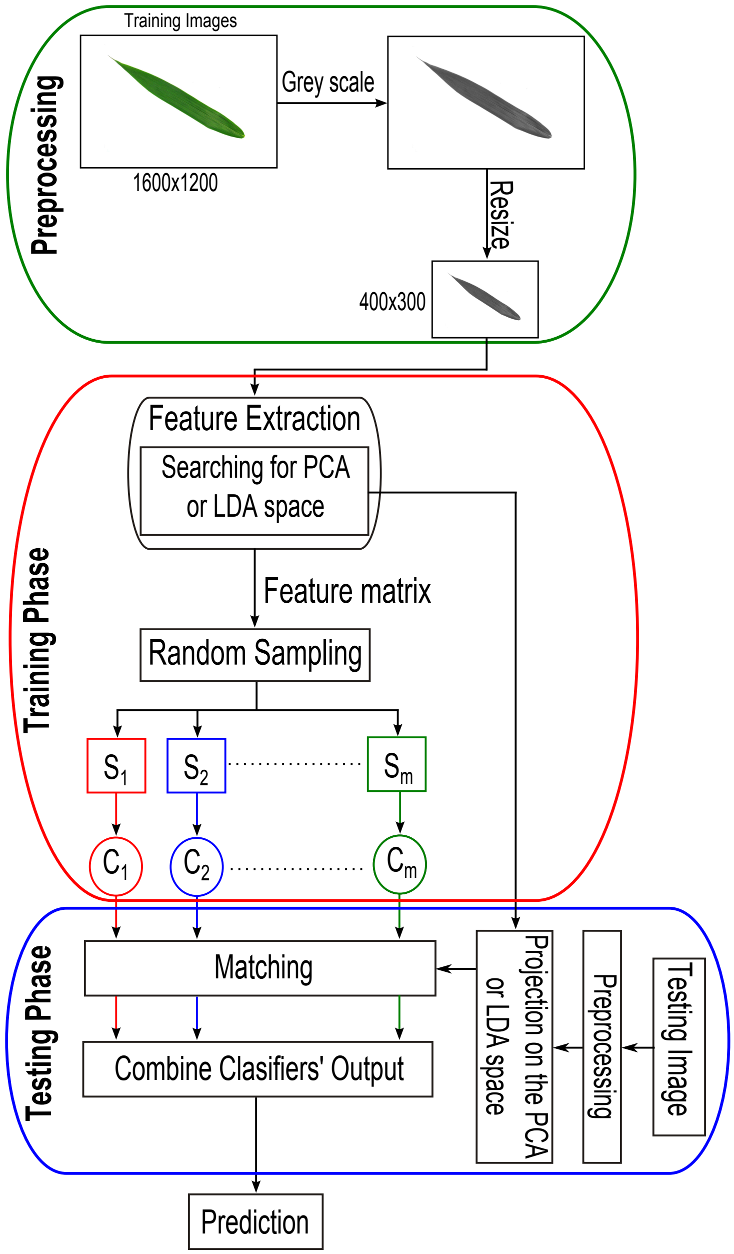


Figure 5: Plant identification system using leaves' images

classifier's model that has been also built in the training phase. Detailed steps of this phase are given in Algorithm (5)

Algorithm 5 : Testing Phase

1: Read an unknown leave image (I_{test}).
2: **if** (1D-based method)) **then**
3: Convert this image $I_{test}(r \times c)$ into a vector form, $I_{test}^{'}((r \times c) \times 1)$.
4: **else**
5: Deals with the image in 2D form (i.e. matrix representation).
6: **end if**
7: Project the unknown 2D image on the projection space to get y_{test}.
8: Match between y_{test} with Y using the Bagging model that built during the training phase to find the class label of the unknown image.

5. Experimental Results

To evaluate our proposed approach, the Flavia public dataset was used. This dataset consists of 1907 colored leaves images with size (1600×1200) and collected from 33 different species. The selected images are in different orientations, illumination, and quality. In this paper, all colored images were converted into grey scale images as shown in Figure 5. Next, all images were resized to be 400×300 to reduce the computational time. Figure 6 shows different samples from the dataset.

Four scenarios were designed to evaluate the performance and accuracy of the proposed model (using 1DPCA, PCA+LDA, DLDA, 2DPCA, and 2DLDA). In these scenarios, the Bagging classifier ensemble, with different numbers of weak learners was used to match the unknown image with the trained images. Due to the high dimensionality of the data, 1DLDA was not suitable for the feature extraction. The reason of this high-dimensionality of the one-dimensional form of the image was $d = 400 \times 300 = 120000$ and hence $d >> M - C$ which leads to SSS problem, where M is the total number of samples and C is the number of classes. To avoid this problem, PCA+LDA and Direct LDA (DLDA) methods were used for the feature extraction in the one-dimensional method.

In the first scenario, the accuracy of the two methods (1D-based and 2D-

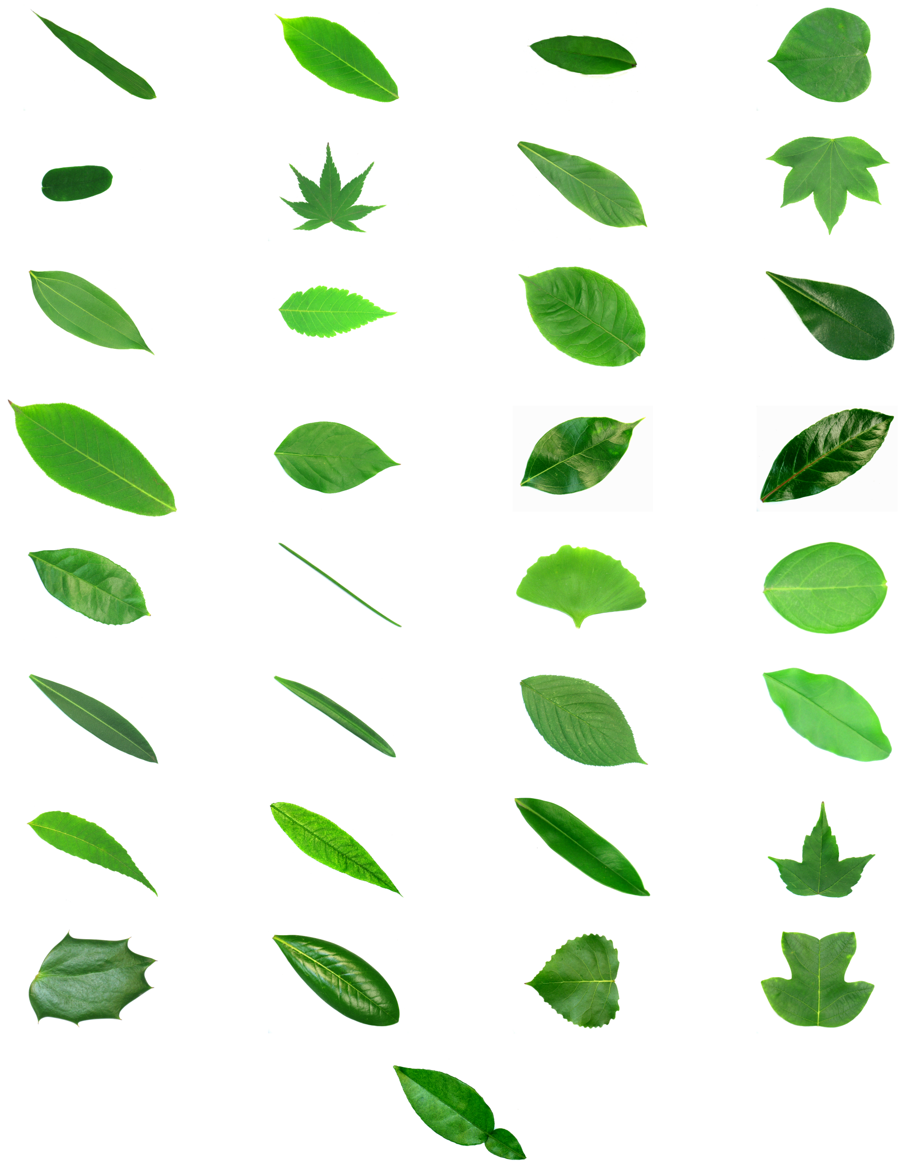
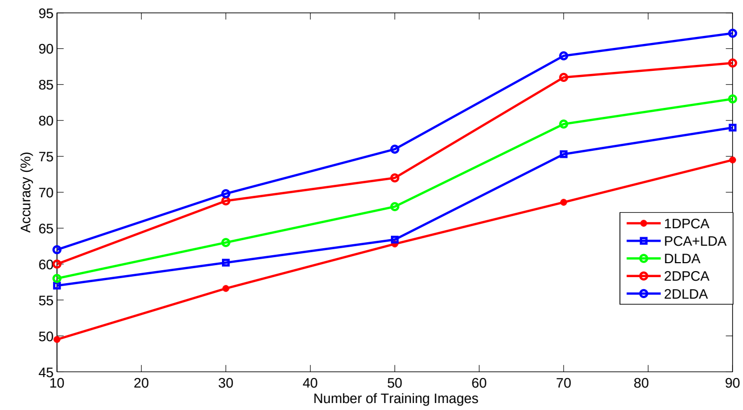


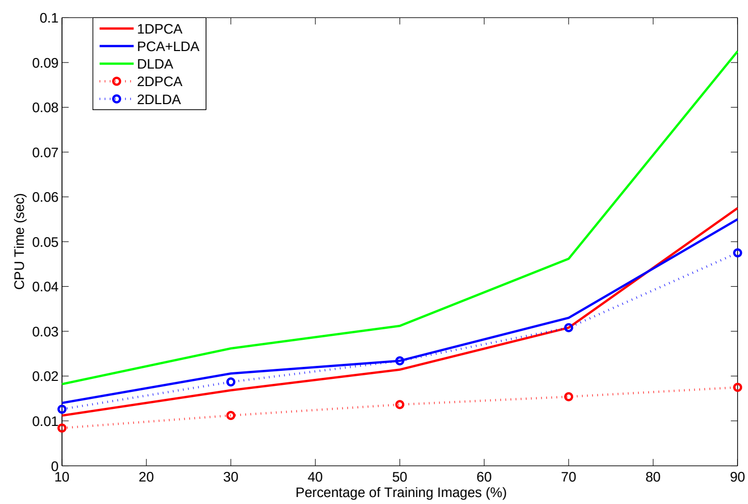
Figure 6: Sample of different leaves' images (one sample from each class or plant).

based) was investigated through testing different percentages of training images of each plant type, i.e. class. The training images were selected randomly from the database while the remaining images, were used during the testing phase.

In this scenario, the size of Bagging classifier was five. The accuracy and CPU time of this scenario are shown in Figure 7.



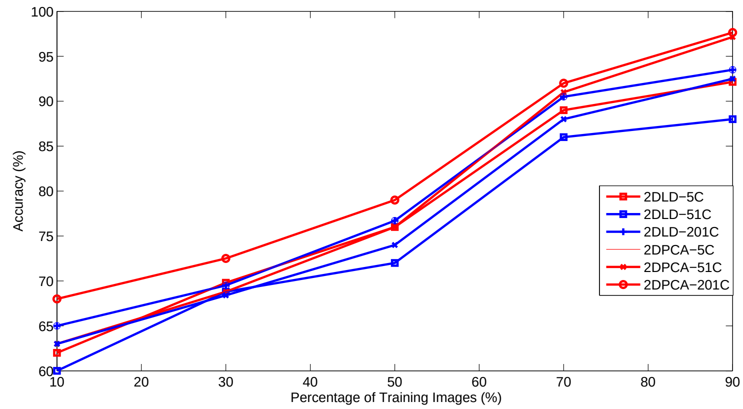
(a)



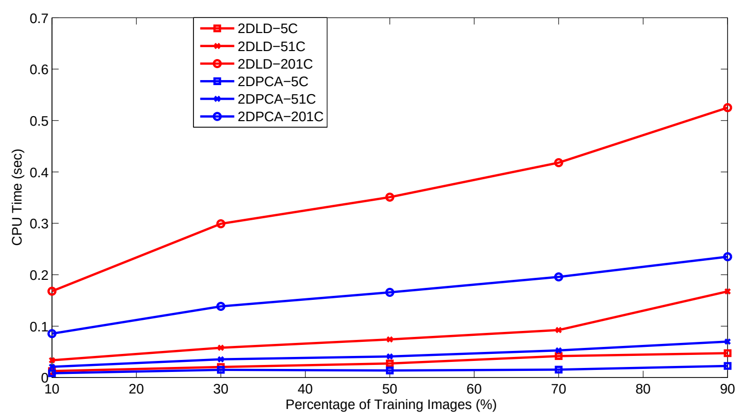
(b)

Figure 7: Accuracy and CPU time of the proposed model using 1DPCA, PCA+LDA, DLDA, 2DPCA, and 2DLDA with different percentages of the training images and five weak learners of the Bagging classifier; (a) Accuracy, (b) CPU time.

The second scenario was designed based on the results of the first one in



(a)



(b)

Figure 8: Accuracy and CPU time of the 2D-based method with different number of training images and weak learners of the Bagging classifier.

which the 2D-based methods gave better results than that of the 1D-based one. Thus, the aim of this scenario was to further understand the effect of changing the number of training images and to evaluate the accuracy and the performance stability over the standardize data. In this scenario, the 2DPCA and 2DLDA were used to extract the images' features. The Bagging classifier was then used in many experiments at different values of its weak learners (i.e. 5, 51, and

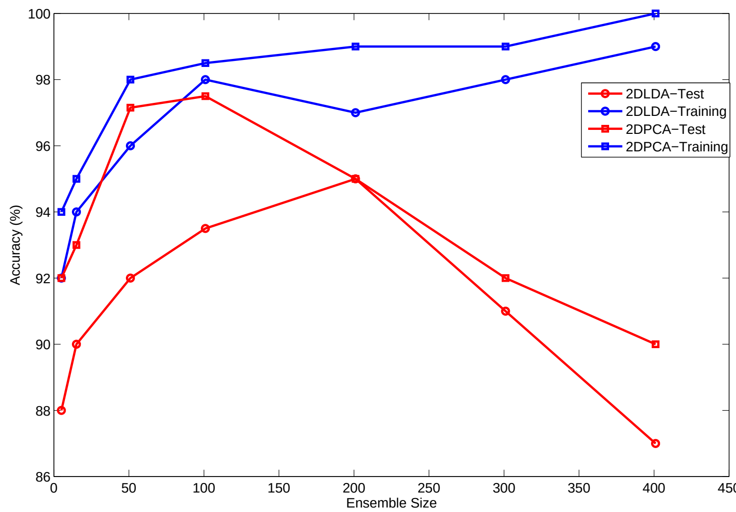
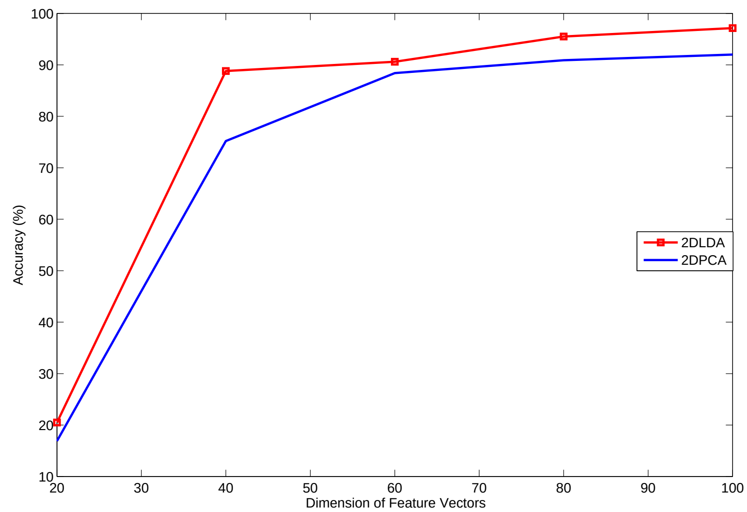


Figure 9: A comparison between the training and testing accuracy of 2DLDA and 2DPCA method using different ensemble sizes.

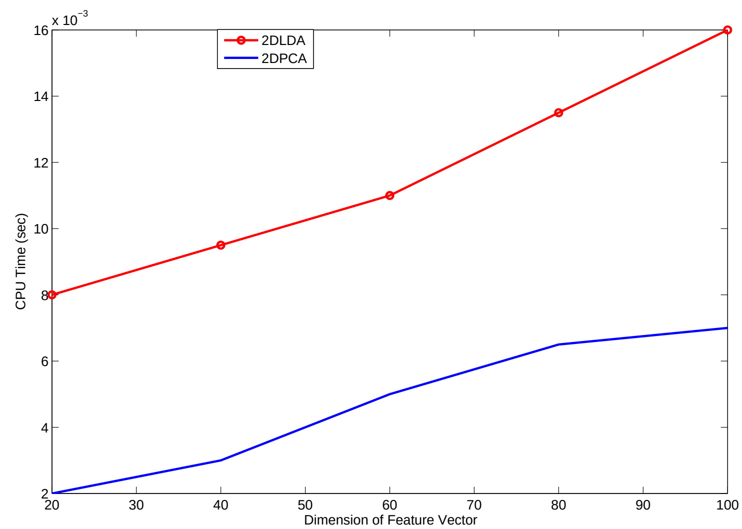
201). In addition, the percentage of training images was ranged from 10% to 90%. The results obtained from this scenario are shown in Figure 8. Moreover, a comparison between the training and testing accuracy of the Bagging model is shown in Figure 9.

The third scenario was conducted to investigate the relationship between the accuracy and the dimension of the feature vectors of the 2DPCA and 2DLDA methods. In other words, the accuracy of the 2DPCA and 2DLDA was tested against different numbers of eigenvectors constructing the projection space. In this experiment, series of different dimensions were used. Moreover, 90% of the images from each class were used to train the model, while the other images were used to test the model. In addition, there were 51 weak learners in the Bagging classifier. Figure 10 shows the results of this experiment.

The fourth and last scenario was conducted to compare the accuracy of the 2DLDA method when different classifiers (Bagging, k-NN, and MLP) were used. In all experiments of this scenario, 51 weak learners were used in the Bagging



(a)



(b)

Figure 10: Accuracy of the two-dimensional methods (2DPCA and 2DLDA) with varying dimensions of the feature vectors; (a) Accuracy, (b) CPU Time.

Table 1: Accuracy rate of the proposed model using Bagging, k-NN, and MLP classifiers.

Class	**Bagging**	**k-NN**	**MLP**	**Class**	**Bagging**	**k-NN**	**MLP**	**Class**	**Bagging**	**k-NN**	**MLP**
1	98	94	98	**12**	100	94	92	**23**	98	94	96
2	100	90	98	**13**	100	86	94	**24**	100	90	96
3	97	90	95	**14**	96	86	92	**25**	100	92	94
4	100	94	96	**15**	98	92	92	**26**	98	94	94
5	96	96	96	**16**	92	90	92	**27**	98	94	96
6	98	90	92	**17**	96	92	92	**28**	98	90	96
7	100	92	88	**18**	96	82	88	**29**	100	92	97
8	96	85	90	**19**	98	94	96	**30**	98	90	94
9	94	88	86	**20**	96	92	96	**31**	92	87	92
10	98	92	94	**21**	92	87	90	**32**	98	86	92
11	98	84	92	**22**	86	81	82	**33**	100	96	96

classifier, five nearest neighbours ($k = 5$) in the k-NN classifier, and 30 and 33 nodes for the hidden and output layers, respectively, in the MLP classifier. Moreover, 90% of the images from each class were used to train the model, while the other images were used to test the model. The accuracy of each class of this experiment are summarized in Table 1

6. Discussion

From the results of the first scenario, shown in Figure 7, the following remarks can be drawn. Firstly, in terms of accuracy issues, the accuracy of all five variants (i.e. 1DPCA, PCA+LDA, DLDA, 2DPCA, and 2DLDA) was improved when the number of training images was increased. This can be explained, as reported in (Brain et al., 1999), using more training images will decrease the variance[4] and hence decreases the overfitting. Secondly, the accuracy of the 2D-based methods (i.e. 2DPCA and 2DLDA) was better than that of the 1D-based methods (i.e. 1DPCA, PCA+LDA, and DLDA). Thirdly, the 2DLDA method achieved the best accuracy and the 1DPCA-based one accomplished the worst accuracy. Fourthly, DLDA method achieved accuracy better than

[4]The variance is the error from sensitivity to small variations in training samples

PCA+LDA method because PCA+LDA method loses more information than DLDA as mentioned in Section 3.1.2.

In terms of the CPU performance, from Figure 7b, it can be noticed that the 2DPCA is the most efficient algorithm among all other methods and the DLDA is the worst one. This can be explained as the high dimensionality of the one-dimensional data. Mathematical interpretation of this point shows that the size of the image covariance matrix using 2DPCA ($c \times c$) is much smaller than in 1DPCA ($(r \times c) \times (r \times c)$). As a result, less time is required to determine the corresponding eigenvectors when the 2DPCA is used. For example, in our case, the size of the image after resizing it was 400×300. Hence, to calculate the covariance matrix of 2DPCA, it is required to multiply two matrices of (300×300). But, when using the 1DPCA, all training images are converted into one vector (1×120000), and the covariance matrix is computed by multiplying two matrices $(M \times 120000) \times (120000 \times M)$, where M represents the total number of training images. Thus, 2DPCA method takes CPU time much lower than 1DPCA method. Similarly, 2DLDA involves the eigen-decomposition of matrix S_W and S_B which have dimensions much smaller than in 1DLDA method. This reduction dramatically reduces the computational time and memory space of 2DLDA method (Ye et al., 2004). Moreover, in 1DLDA, S_W is singular in most cases because the dimension of the samples is greater than the number of samples in each class. However, 2DLDA overcome this problem efficiently because the rank of any training image is equal to $min(r, c)$. Hence, the rank of S_W is less than or equal to $(M - C).min(r, c)$ (Li and Yuan, 2005). Thus, in 2DLDA, S_W is nonsingular when Equation (9) is true. In real practical problems, Equation (9) is always satisfied. Thus, S_W is always nonsingular, hence, SSS problem can be solved using 2DLDA (Li and Yuan, 2005).

$$M \geq C + \frac{c}{min(r, c)} \tag{9}$$

From Figure 8 the following remarks can be noticed. Firstly, the higher number of iterations of Bagging classifier used, the better classification accuracy achieved. However, this was accomplished on the cost of taking more CPU time (see Figure 8b). Secondly, the 2DLDA method achieved identification accuracy better than that of the 2DPCA method, but this was also accomplished with more CPU time. This is because of LDA searches in the space that extracts the most discriminative features, while the PCA searches in the space that extracts the data with the high variance. Thirdly, increasing the ensemble size led to the complexity of the bagging model and hence took more CPU time and may lead to the overfitting problem. Figure 9 shows a comparison between the training and testing accuracy. In this figure, the training accuracy of 2DLDA and 2DPCA methods was increased till it reached to an extent at which it remained constant. On the other hand, the testing accuracy was increased when the ensemble size was increased till it reached to an extent after which it reduced again. As shown in the figure, the best ensemble size was approximately 201.

From Figure 10a, two remarks can be noticed. First, the accuracy of the 2DPCA and 2DLDA methods is proportional with the number of eigenvectors. Second, a major change (about 60%) in the accuracy achieved when the percentage of the eigenvectors was increased from 20% to 40%. But, a minor change (about 5%) in the accuracy achieved when the percentage of the accuracy ranged from 40% to 100 %. This means that the most discriminative feature are concentrated nearly in the first half of the eigenvectors. In terms of CPU time and from Figure 10b, it is clear the computational time of the 2DPCA and 2DLDA methods increased when more eigenvectors were used to construct the PCA or LDA space.

From Table 1, two remarks can be seen. First, the Bagging classifier achieved the best accuracy rate (97.15%), while MLP and k-NN classifiers achieved 93.15% and 90.18%, respectively. The accuracy of the classes was ranged from 86% to 100% when Bagging classifier was used.

To further evaluate our proposed approach (2DPCA and 2DLDA which gave the best results), a comparison was conducted with some state-of-the-art approaches which used different feature extraction methods and classifiers for the same dataset. The results of this comparison are shown in Table 2. From this

Table 2: A comparison between our proposed plant identification method and some of state-of-the-art methods in terms of, classification accuracy, size of database images, feature extraction methods.

Author	**Feature Extraction Method**	**Classification Method**	**Database Images**	**Results**
(Arun Priya et al., 2012b)	Digital Morphological Features (DMFs) + PCA	k-NN SVM	5 classes (331 images)	k-NN (78%) SVM (94.5%)
(Caglayan et al., 2013)	Color+Shape	k-NN SVM NB RF	32 classes (1897 images)	k-NN (94.2%) SVM (92.9%) NB (88.95%) RF (96.32%)
(Satti et al., 2013)	Color+Shape	k-NN ANN	33 classes (1907 images)	k-NN (85.9%) ANN (93.3%)
(Chaki et al., 2015)	Texture+Shape	NFC MLP	31 classes (930 images)	NFC (81.6%) MLP (87.1%)
(Chaki et al., 2016)	Shape+Texture (statistical)	NFC	30 class (600 images)	NFC (97%)
Proposed Model	1DPCA, DLDA, PCA+LDA, 2DPCA, 2DLDA	Bagging	33 classes (1907 images)	1DPCA (72%) PCA+LDA (77%) DLDA (82%) 2DPCA (93.5%) 2DLDA (97.12%)

table, the following remarks can be drawn. Firstly, although the proposed approach and the one proposed by Satti *et al.* used all the classes of the Flavia dataset (i.e, 33 classes), while the other approaches excluded some classes, our proposed approach achieved the highest accuracy (97.12%). Secondly, Chaki *et al.* also achieved high accuracy at (97%), but they used only 30 classes and 600 images while in our approach 33 classes and 1907 images were used in all experiments.

As a general remark, from Figure 7 and Figure 8, it can be noticed that the accuracy of the proposed approach with its variants is proportional to the number of training images and the best accuracy is achieved when 90% of the training images is used.

7. Conclusion

This paper presented a plant identification approach based on their 2D leaves images. The approach consists of two main phases: feature extraction and classification. In the first phase, five algorithms (1DPCA, 1DLDA, Direct-LDA, PCA+LDA, 2DPCA, and 2DLDA) were applied to extract the leaves features. In the second phase, the Bagging classifier was employed to test which feature extraction technique could give the best accuracy and performance. The five variants of the proposed approach were evaluated using all leave images of Flavia dataset. The evaluation results showed the variants used the 2DPCA and 2DLDA were much better than the ones used the PCA, PCA+LDA, and Direct-LDA. It also was found that the 2DLDA-based method was the best one. In addition, experiments conducted for the Bagging classifier parameter (the size of the weak learners) proved that the classification accuracy increased when this parameter increased. Moreover, the results showed that the classification accuracy of the 2DPCA and 2DLDA methods was proportional with the number of the selected eigenvectors and the highest accuracy was (97.12%) and achieved using 2DLDA. Last but not least, a comparison with the most related work showed that our approach achieved better accuracy under the same dataset and same experimental setup. In the future work, deep learning techniques will be investigated for plant identification using the same leaves' dataset.

Arora, A., Gupta, A., Bagmar, N., Mishra, S., Bhattacharya, A., 2012. A plant identification system using shape and morphological features on segmented leaflets: Team iitk, CLEF 2012. In: CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012. URL http://ceur-ws.org/Vol-1178/CLEF2012wn-ImageCLEF-AroraEt2012.pdf

Arun Priya, C., Balasaravanan, T., Thanamani, A. S., 2012a. An efficient leaf recognition algorithm for plant classification using support vector machine. In: Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on. IEEE, pp. 428–432.

Arun Priya, C., Balasaravanan, T., Thanamani, A. S., 2012b. An efficient leaf recognition algorithm for plant classification using support vector machine. In: International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME). IEEE, pp. 428–432.

Brain, D., Webb, G., Richards, D., Beydoun, G., Hoffmann, A., Compton, P., 1999. On the effect of data set size on bias and variance in classification learning. In: Proceedings of the Fourth Australian Knowledge Acquisition Workshop, University of New South Wales. pp. 117–128.

Caglayan, A., Guclu, O., Can, A. B., 2013. A plant recognition approach using shape and color features in leaf images. In: International Conference on Image Analysis and Processing (ICIAP). Springer, pp. 161–170.

Chaki, J., Parekh, R., 2012. Plant leaf recognition using gabor filter. International Journal of Computer Applications 56 (10).

Chaki, J., Parekh, R., Bhattacharya, S., 2015. Plant leaf recognition using texture and shape features with neural classifiers. Pattern Recognition Letters 58, 61–68.

Chaki, J., Parekh, R., Bhattacharya, S., 2016. Plant leaf recognition using ridge filter and curvelet transform with neuro-fuzzy classifier. In: Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics. Springer, pp. 37–44.

Feng, T.-t., Wu, G., 2014. A theoretical contribution to the fast implementation of null linear discriminant analysis method using random matrix multiplication with scatter matrices. arXiv preprint arXiv:1409.2579.

Gaber, T., Tharwat, A., Snasel, V., Hassanien, A. E., 2015. Plant identification: Two dimensional-based vs. one dimensional-based feature extraction methods. In: 10th International Conference on Soft Computing Models in Industrial and Environmental Applications. Springer, pp. 375–385.

Galdámez, P. L., Arrieta, A. G., Ramón, M. R., 2015. A small look at the ear recognition process using a hybrid approach. Journal of Applied Logic.

Kuncheva, L. I., 2014. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, Second Edition.

Li, M., Yuan, B., 2005. 2d-lda: A statistical linear discriminant analysis for image matrix. Pattern Recognition Letters 26 (5), 527–532.

Lowe, D. G., 1999. Object recognition from local scale-invariant features. In: The proceedings of the seventh IEEE international conference on Computer vision, 1999. Vol. 2. Ieee, pp. 1150–1157.

Lu, J., Plataniotis, K. N., Venetsanopoulos, A. N., 2005. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. Pattern Recognition Letters 26 (2), 181–191.

Marcialis, G. L., Roli, F., 2002. Fusion of lda and pca for face verification. In: Biometric Authentication. Springer, pp. 30–37.

Mitchell, J. B., 2014. Machine learning methods in chemoinformatics. Wiley Interdisciplinary Reviews: Computational Molecular Science 4 (5), 468–481.

Moore, B., 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. IEEE Transactions on Automatic Control 26 (1), 17–32.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7), 971–987.

Salvador, E., Cavallaro, A., Ebrahimi, T., 2004. Cast shadow segmentation using invariant color features. Computer vision and image understanding 95 (2), 238–259.

Satti, V., Satya, A., Sharma, S., 2013. An automatic leaf recognition system for plant identification using machine vision technology. International Journal of Engineering Science and Technology (IJEST) ISSN, 0975–5462.

Strang, G., 2003. Introduction to linear algebra. Wellesley-Cambridge Press, Massachusetts, Fourth Edition.

Tharwat, A., Gaber, T., Hassanien, A. E., 2014a. Advanced Machine Learning Technologies and Applications: Second International Conference, AMLTA 2014, Cairo, Egypt, November 28-30, 2014. Proceedings. Springer International Publishing, Cham, Ch. Cattle Identification Based on Muzzle Images Using Gabor Features and SVM Classifier, pp. 236–247.
URL http://dx.doi.org/10.1007/978-3-319-13461-1_23

Tharwat, A., Gaber, T., Hassanien, A. E., Hassanien, H. A., Tolba, M. F., 2014b. Cattle identification using muzzle print images based on texture features approach. In: Proceedings of the 5th International Conference on Innovations in Bio-Inspired Computing and Applications, IBICA, June 23-25, 2014, Ostrava, Czech. Springer, pp. 217–227.

Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M., Refaat, B., 2015. Sift-based arabic sign language recognition system. In: Afro-European Conference for Industrial Advancement. Vol. 334. Springer, pp. 359–370.

Tharwat, A., Ibrahim, A., Ali, H., 2012. Personal identification using ear images based on fast and accurate principal component analysis. In: 8th International Conference on Informatics and Systems (INFOS). IEEE, pp. 56–59.

Turk, M. A., Pentland, A. P., 1991. Face recognition using eigenfaces. In: Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'91. IEEE, pp. 586–591.

Uluturk, C., Ugur, A., 2012. Recognition of leaves based on morphological features derived from two half-regions. In: International Symposium on Innovations in Intelligent Systems and Applications (INISTA). IEEE, pp. 1–4.

Valliammal, N., Geethalakshmi, S., 2011. Automatic recognition system using preferential image segmentation for leaf and flower images. An International Journal of Computer Science & Engineering (CSEIJ) 1 (4), 13–25.

Welling, M., 2005. Fisher linear discriminant analysis. Department of Computer Science, University of Toronto.

Wu, M. C., Zhang, L., Wang, Z., Christiani, D. C., Lin, X., 2009. Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. Bioinformatics 25 (9), 1145–1151.

Yang, J., Zhang, D., Frangi, A. F., Yang, J.-y., 2004. Two-dimensional pca: a new approach to appearance-based face representation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1), 131–137.

Ye, J., Janardan, R., Li, Q., 2004. Two-dimensional linear discriminant analysis. In: Neural Information Processing Systems, NIPS, December 13-18, 2004, Vancouver, British Columbia, Canada]. pp. 1569–1576.

Ye, J., Xiong, T., 2006. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. The Journal of Machine Learning Research 7, 1183–1204.